



Teknillinen korkeakoulu  
Graafisen tekniikan laboratorio

Helsinki University of Technology  
Laboratory of Graphic Arts Technology

---

# Hajautetun painatuksen kapasiteetti

Pontus Hellman

Tämä diplomityö on jätetty opinnäytteenä tarkastettavaksi  
diplomi-insinöörin tutkintoa varten Espoossa 9. marraskuuta 1995

Työn valvoja: professori Hannu Saarelma  
Työn ohjaaja: professori Hannu Saarelma

Tekijä:

Pontus Hellman

Työn nimi:

**Hajautetun painatuksen kapasiteetti**

Päivämäärä:

09.11.1995

Sivumäärä:

115

Osasto:

Prosessi- ja materiaalitekniiikan osasto

Professuuri:

Aut-75 Graafinen tekniikka

Työn valvoja:

Professori Hannu Saarelma

Työn ohjaaja:

Professori Hannu Saarelma

Elektronisille painatusjärjestelmille on luonteenomaista tuotannon nopeatempoisuus ja pienet painokset. Tämän tyyppinen tuotanto on tehokasta ainoastaan siinä tapauksessa, että tuotantoa pystytään ohjaamaan tehokkaasti. Onnistuneen tuotannonohjauksen tärkein edellytys taas on, että järjestelmän suorituskykyä pystytään ennustamaan. Tällä hetkellä elektroniset painatusjärjestelmät rakentuvat suurtehotulostimien ympärille eli voidaan puhua keskitetystä tulostamisesta. Oletettavaa kuitenkin on, että jo lähitulevaisuudessa tulostusta hajautetaan yhä lähemmäksi varsinaista loppukäyttäjää, jolloin suorituskyvyn ennustaminen on huomattavasti vaikeampaa. Tämän diplomityön tavoitteena on sellaisten mallien kehittäminen, joiden avulla tietokoneverkkojen kautta hajautettujen elektronisten painatusjärjestelmien kokonaissuorituskykyä pystyttäisiin ennustamaan.

Kirjallisuusosassa on analysoitu hajautettujen elektronisten painatusjärjestelmien yleistä rakennetta ja suorituskykyyn vaikuttavia tekijöitä. Erityisesti on paneuduttu eri tiedonsiirtotekniikoihin, koska niiden ominaisuuksilla on merkittävä vaikutus hajautettujen järjestelmien suorituskykyyn. Koska aiemmissa tutkimuksissa on käynyt ilmi, että elektronisten painatusjärjestelmien mallintaminen perinteisillä menetelmillä ei onnistu, tässä työssä hyödynnettiin niin sanottuja oppivia ja älykkäitä menetelmiä. Näistä valittiin mallinnuksessa käytettäväksi neurolaskenta. Varsinainen mallinnus jaettiin kahteen osaan. Ensin mallinnettiin järjestelmään kuuluvat yksittäiset tulostimet neuroverkkomallien avulla. Samassa yhteydessä selvitettiin, kuinka tarkasti järjestelmää on malleille kuvattava, jotta saadut ennusteet olisivat tarkkoja. Tämän jälkeen yksittäisten tulostimien suorituskykyennusteiden perusteella laskettiin koko järjestelmän suorituskyky.

Yhden tulostimen suorituskyvyn ennustamiseen kehitettyjen mallien ennustustarkkuus ei neuroverkkojen opetukseen käytetyn materiaalin pohjalta muodostunut riittävän tarkaksi. Mallien tekemät virheet ovat keskimäärin 20-30% luokkaa. Käytetty opetusaineisto ei ollut riittävän laaja ja monipuolinen, jotta muodostetut neuroverkkoarkkitehtuurit olisivat oppineet suorituskykyyn vaikuttavien tekijöiden väliset riippuvuudet. Tästä syystä järjestelmän kokonaissuorituskykyä ennustavia malleja testattaessa päädyttiin käyttämään mitattuja, ei ennustettuja tulostusaikoja. Tällä tavoin varmistettiin, että neuroverkkomallien avulla laskettujen ennusteiden virheet eivät vääristä kokonaissuorituskykymallien ennusteita. Järjestelmän kokonaissuorituskyvyn ennustamiseen kehitetyt mallit olivat erittäin tarkkoja. Tulostusarkkitehtuurien mallintamiseen kehitettyjen algoritmien tekemät virheet ovat keskimäärin alle prosentin luokkaa. Tästä voidaan päätellä, että tulostusjärjestelmän eri osien väliset riippuvuudet ovat riittävällä tarkkuudella lineaarisia.

Positiivista tuloksissa on, että neuroverkkomallien virhelähteet ovat yleensä helposti osoitettavissa ja kaikki viittaa siihen, että tulostimien mallintaminen on riittävän laajalla ja monipuolisella opetusaineistolla mahdollista. Jotta neuroverkkojen opetukseen tarvittavaa materiaalia saataisiin riittävästi, tulisi mallit integroida osaksi reaalista järjestelmää, jolloin informaatiota voitaisiin kerätä automaattisesti ja malleja opettaa koko ajan. Kokonaisen tulostusjärjestelmän suorituskyvyn ennustaminen ei enää tämän jälkeen ole ongelma. Kaikki järjestelmäkohtaiset ominaisuudet, tulostimien ominaisuuksia lukuunottamatta voidaan mallintaa yksinkertaista matematiikkaa soveltamalla eli riippuvuudet ovat lineaarisia ja vähäisellä vaivalla testitulostusten avulla selvitettävissä.



# SISÄLLYSLUETTELO

	<b>KÄYTETYT LYHENTEET .....</b>	<b>v</b>
	<b>ESIPUHE .....</b>	<b>vii</b>
<b>1</b>	<b>JOHDANTO .....</b>	<b>8</b>
<b>2</b>	<b>HAJAUTETUT PAINATUSJÄRJESTELMÄT .....</b>	<b>10</b>
	2.1 Määritelmä .....	10
	2.2 Kapasiteettiin vaikuttavat tekijät .....	12
	2.2.1 Hajautetun tulostusjärjestelmän komponentit .....	12
	2.2.2 Tulostettavan materiaalin rakenne ja sisältö .....	13
	2.3 Postscript-sivunkuvauskieli .....	15
<b>3</b>	<b>HAJAUTETUN PAINATUSJÄRJESTELMÄN TEKNINEN TOTEUTUS .....</b>	<b>17</b>
	3.1 Yleistä tiedonsiirtoverkoista.....	17
	3.1.1 Peruskäsitteitä .....	17
	3.1.2 Siirtomediat.....	18
	3.1.3 Verkkojen luokittelu .....	20
	3.2 Lähiverkot (LAN) ja alueverkot (MAN).....	21
	3.2.1 Verkkotopologiat .....	21
	3.2.2 Lähiverkkotekniikat .....	23
	3.2.3 Lähiverkkojen yhdistäminen.....	25
	3.3 Suuralueverkot (WAN).....	26
	3.3.1 ISDN .....	26
	3.3.2 ATM.....	27
	3.4 Palvelimet.....	28
	3.4.1 Tiedostopalvelimet .....	28
	3.4.2 Tulostinpalvelimet.....	29
	3.5 Tulostimet .....	30
	3.5.1 NIP-teknikat .....	30
	3.5.2 Tulostimien luokittelu .....	32
	3.6 Rasterointiprosessori .....	33
<b>4</b>	<b>MALLINNUSTEKNIIKAT .....</b>	<b>35</b>
	4.1 Oppivat ja älykkäät järjestelmät .....	35
	4.1.1 Määritelmä .....	35
	4.1.2 Yleisimmät menetelmät.....	35
	4.1.3 Menetelmien vertailu.....	37
	4.2 Neurolaskenta .....	38
	4.2.1 Periaatteet ja matemaattinen tausta .....	38
	4.2.2 Neurolaskennan menetelmien luokittelu .....	41
	4.2.3 Tunnetuimmat neurolaskennan menetelmät .....	43
	4.2.4 Neurolaskennan soveltaminen käytännössä.....	44
<b>5</b>	<b>KOESUUNNITELMA .....</b>	<b>46</b>
	5.1 Käytetyn materiaalin rakenne.....	46
	5.1.1 Opetusaineisto .....	46
	5.1.2 Testiaineisto .....	47
	5.2 Koeympäristö .....	47
	5.3 Suoritettavien kokeiden kuvaus.....	50
<b>6</b>	<b>NEUROLASKENNAN SOVELTAMINEN MALLINNUKSESSA .....</b>	<b>52</b>
	6.1 Opetusmateriaalin analysointi.....	52
	6.2 Neuroverkkomallinnus käytännössä.....	60

	6.2.1 Syöte- ja vasteparametrien käsittely .....	60
	6.2.2 Käytettävän menetelmän valitseminen .....	61
	6.2.3 Oppimisalgoritmin valitseminen .....	61
	6.2.4 Neuroverkkoarkkitehtuurin optimointi.....	62
<b>7</b>	<b>YHDEN TULOSTIMEN SUORITUSKYKYMALLIT .....</b>	<b>63</b>
	7.1 Mallintamisen perusteet .....	63
	7.2 Testitulostimien suorituskymallit .....	64
	7.3 Käsitellyt erikoistapaukset.....	69
	7.4 Yleinen yhden tulostimen malli .....	70
<b>8</b>	<b>HAJAUTETTUIJEN PAINATUSJÄRJESTELMIEN SUORITUSKYKYMALLIT .....</b>	<b>73</b>
	8.1 Hajautettujen järjestelmien mallintaminen .....	73
	8.1.1 Mallintamisen lähtökohta .....	73
	8.1.2 Tulostustapahtuman analysointi.....	74
	8.1.3 Käytettyjen työasemien ominaisuuksien määrittäminen .....	75
	8.1.4 Yleistä kehitetyistä algoritmeista.....	76
	8.2 "Pipeline"-simulaatio .....	76
	8.2.1 Järjestelmäanalyysi.....	76
	8.2.2 Algoritmien kuvaus.....	78
	8.2.3 Kehitettyjen algoritmien testaus .....	79
	8.3 Yhden työaseman ja useamman tulostimen ympäristöt .....	81
	8.3.1 Järjestelmäanalyysi.....	81
	8.3.2 Algoritmien kuvaus.....	82
	8.3.3 Kehitettyjen algoritmien testaus .....	83
	8.4 Useamman työaseman ja useamman tulostimen ympäristöt .....	84
	8.4.1 Järjestelmäanalyysi.....	84
	8.4.2 Algoritmien kuvaus.....	85
	8.4.3 Kehitettyjen algoritmien testaus .....	85
<b>9</b>	<b>MALLIEN SOVELTAMINEN KÄYTÄNNÖSSÄ.....</b>	<b>87</b>
	9.1 Mallien hyödyntäminen nykyisessä muodossaan .....	87
	9.2 Laajamittaisen soveltamisen edellytykset .....	88
	9.3 Ongelmat .....	90
	9.3.1 Syöteparametrien selvittäminen tiedostoista .....	90
	9.3.2 Verkkokuorman huomioiminen.....	90
<b>10</b>	<b>MAHDOLLINEN JATKOKEHITYS.....</b>	<b>91</b>
<b>11</b>	<b>JOHTOPÄÄTÖKSET .....</b>	<b>93</b>
<b>12</b>	<b>YHTEENVETO .....</b>	<b>96</b>
	<b>LÄHDELUETTELO.....</b>	<b>100</b>
	<b>LIITE 1: OPETUSMATERIAALIN RAKENNE JA DATAMÄÄRÄT .....</b>	<b>103</b>
	<b>LIITE 2: OPETUSMATERIAALIN RAKENNE (%-OSUUDET) .....</b>	<b>104</b>
	<b>LIITE 3: TESTIMATERIAALIN RAKENNE .....</b>	<b>105</b>
	<b>LIITE 4: OPETUSMATERIAALIN SIIRTO- JA TULOSTUSAJAT .....</b>	<b>106</b>
	<b>LIITE 5: HAJAUTETTUIJEN JÄRJESTELMIEN TESTIEN TULOKSET .....</b>	<b>107</b>
	<b>LIITE 6: SUORITUSKYKYALGORITMIEN C-LÄHDEKODI .....</b>	<b>112</b>



# KÄYTETYT LYHENTEET

AFP .....	AppleTalk Filing Protocol
ANSI .....	American National Standards Institute
ASCII .....	American Standard Code of Information Interchange
ATM .....	Asynchronous Transfer Mode
B-ISDN .....	Broadband Integrated Services Digital Network
bit/s .....	bittä sekunnissa
BMP .....	Bitmap
BP .....	Backpropagation
CD-ROM .....	Compact Disk-Read Only Memory
CGM .....	Computer Graphics Metafile
COM .....	Sarjaportti
CPU .....	Central Processing Unit
CS .....	Continuous Stream
CSMA/CD .....	Carrier Sense Multiple Access with Collision Detection
DOD .....	Drop-On-Demand
dpi .....	dots per inch
DSC .....	Document Structuring Conventions
EISA .....	Extended Industry Standard Architecture
EPS .....	Encapsulated PostScript
FDDI .....	Fiber Distributed Data Interface
GKS .....	Graphical Kernel System
Gt .....	Gigatavu
HPGL .....	Hewlett-Packard Graphics Language
HP .....	Hewlett-Packard
h .....	tunti
Hz .....	Hertsi
I/O .....	input/output
IBM .....	International Business Machines
IEEE .....	(The) Institute of Electrical and Electronics Engineers
IFD .....	Image File Directory
IPX .....	Internetwork Packet Exchange
ISDN .....	Integrated Services Digital Network
ITU .....	International Telecommunication Union
kbit/s .....	kilobittä sekunnissa
kHz .....	kilohertsi
km .....	kilometri
kpl .....	kappaletta
l" .....	linjaa tuumalla
l/cm .....	linjaa senttimetrillä
LAN .....	Local Area Network
LPT .....	Rinnakkaisportti
MAC .....	Medium Access Control

---

MAN .....	Metropolitan Area Network
Mbit/s .....	megabittiä sekunnissa
MHz .....	megahertsi
min .....	minuutti
mk .....	markka
MLP .....	Multi-layer Perceptron
m .....	metri
Mmk .....	miljoonaa markkaa
µm .....	mikrometri
Mt .....	megatavu
NI .....	Non-Impact
NIP .....	Non-Impact Printing
N-ISDN .....	Narrowband Integrated Services Digital Network
PCI .....	Peripheral Component Interconnect
PCL .....	Printer Control Language
PC .....	Personal Computer
PDF .....	Portable Document Format
p .....	penni
PS .....	PostScript
pt .....	point
R <sup>2</sup> .....	regressiomallin selityssaste
RET .....	Resolution Enhancement Technique
RIP .....	Raster Image Processor
s/min .....	sivua minuutissa
SCSI .....	Small-Computer System Interface
SLB .....	Soft Loadable System
SOM .....	Self-Organizing Map
s .....	sekunti
TCP/IP .....	Transmission Control Protocol/Internet Protocol
T <sub>e</sub> .....	ennustettu tulostusaika
TIFF .....	Tagged Image File Format
T <sub>m</sub> .....	mitattu tulostusaika
WMF .....	Windows Metafile



## ESIPUHE

Tämä diplomityö on laadittu Teknillisen korkeakoulun Graafisen tekniikan laboratoriossa osana TEKESin tavoitetutkimushanketta "Tosiaikainen kirjatuotanto verkkoympäristössä".

Haluaisin kiittää diplomityöni valvojaa ja ohjaajaa professori Hannu Saaremaa erittäin mielenkiintoisesta ja haastavasta aiheesta. Häneltä saamani ohjaus ja ideat olivat työn onnistumisen kannalta korvaamattomia. Lisäksi haluaisin kiittää Teknillisen korkeakoulun Graafisen tekniikan laboratorion koko henkilökuntaa ja erityisesti Oskar Lönnbergiä saamastani avusta. Suuri kiitos kuuluu myös ystäväilleni, jotka pitivät mieleni virkeänä näinä työntäyteisinä kuukausina. Lopuksi haluaisin vielä kiittää vanhempiani, joiden kannustamana aikoinaan akateemiset opintoni aloitin ja viimeisenä, mutta ei vähäisimpänä kihlattuani ja elinkumppaniani Saria, jonka tukea ja pitkämielisyyttä ilman tämä työ ei ikinä olisi nähnyt päivänvaloa.

Pontus Hellman

Espoossa 9. marraskuuta 1995

## 1

## JOHDANTO

*Hajautettu painaminen* on käsitteenä hankala, koska sen sisältö riippuu hyvin pitkälle ympäristöstä jossa toimitaan. Sanomalehtipainoille *hajautettu painaminen* voi olla satelliittipainojen käyttämistä, toimistoissa verkkotulostusjärjestelmä muodostaa hajautetun tulostusympäristön, maailmanlaajuisten tietokoneverkkojen tietokannoista voidaan hakea informaatiota ja tulostaa se paikallisella tulostimella. Jokainen edellä mainituista edustaa tietyn tyyppistä hajautettua painatusjärjestelmää ja esimerkkejä voisi keksiä loputtomasti lisää. Perusperiaate on kuitenkin aina sama: painaminen/tulostaminen tapahtuu fyysisesti eri paikassa kuin informaation käsittely ja varastointi. Informaation siirtäminen voidaan toteuttaa monilla tavoin, kuten levykkeellä, CD-ROM(Compact disk-Read Only Memory)-levyllä, magneettinauhalla, tietokoneverkon kautta ja niin edelleen.

Tässä diplomityössä *hajautetulla painamisella* tarkoitetaan tietokoneverkon kautta tietokannasta *on-demand*ina tapahtuvaa painamiseen rinnastettavaa tulostamista. Yleisempiä nimityksiä tekniikalle ovat *elektroninen painaminen* tai *tietokonepainaminen*. Pääpaino tulee asettaa sanalle tulostaminen, eli konventionaalisia painomenetelmiä ei käsitellä. Lisäksi tulostuksen on oletettu aina tapahtuvan kiinteästi järjestelmään integroidun tietokoneverkon kautta, muita tiedonsiirtotapoja ei ole käsitelty. Työ katsottiin tarpeelliseksi rajata näin, koska muun tyyppiset järjestelmät sisältävät yleensä suuren joukon ei-mittavissa olevia muuttujia, joiden käyttäytymistä ei voida riittävällä tarkkuudella ennustaa. Lisäksi se on perusteltua siitä syystä, että lähes kaikki hajautetut *on-demand*-painatusjärjestelmät tullaan mitä luultavimmin jo lähitulevaisuudessa rakentamaan tietokoneverkkojen ympärille.

*Elektroniseksi painamiseksi* luokiteltava tulostaminen on muutaman viime vuoden aikana lisääntynyt merkittävästi ja kasvun on ennustettu jatkuvan. Tämä johtuu pitkälti siitä, että elektrofotografia on teknologiana kehittynyt ja tietokoneiden suorituskyky on parantunut. Lisäksi tähän ovat vaikuttaneet taloudelliset näkökohdat eli varastointikustannukset pienenevät, pienet painokset ovat kannattavia, toimitusajat lyhenevät jne. Elektronisen painamisen osuus graafisen alan tuotannosta on tällä hetkellä noin 1% eli 150 Mmk. Ennusteissa osuuden on arveltu jopa kymmenkertaistuvan vuoteen 2000 mennessä /3/. *Elektroninen painaminen* on tällä hetkellä kilpailukykyinen vaihtoehto perinteisiin painomenetelmiin verrattuna alle 3000 kappaleen painoksissa, ja kun elektrofotografiatekniikalla saavutettu laatu katsotaan tuotteelle riittäväksi. Tuotteet, joita elektronisilla painojärjestelmillä tällä hetkellä tuotetaan ovat pääasiassa oppikirjoja, manuaaleja ja muita usein päivitettäviä, runsaasti vaihtuvaa informaatiota sisältäviä julkaisuja.

Hajautetun painatusjärjestelmän kokonaissuorituskyvyn muodostumiseen vaikuttavat sen osakomponenttien suorituskyky ja ominaisuudet. Lisäksi tulostettavan materiaalin sisällöllä ja rakenteella, sekä käytettävän sivunkuvauskielen ominaisuuksilla on todettu olevan merkittävä vaikutus tulostusjärjestelmän suorituskykyyn. Graafisen tekniikan laboratoriossa aiemmin laaditusta diplomityöstä kävi ilmi, että elektronisen painatusjärjestelmän mallintaminen perinteisin menetelmin<sup>1</sup> on erittäin vaikeaa ellei jopa mahdotonta. Yksittäisten osien suorituskykyä pystytään ennustamaan riittävällä tarkkuudella, mutta koko

1. Esimerkiksi fysiikkaan perustuvat mallit, asiantuntijajärjestelmät ja tilastolliset menetelmät.



järjestelmän mallintaminen ei onnistu, koska riippuvuudet vaikuttavien tekijöiden välillä eivät ole lineaarisia. Tässä diplomityössä ongelmaa pyritään lähestymään hieman eri tavoin: tavoitteena on rakentaa oppiva ja älykäs malli, jonka avulla olisi mahdollista suorituskykyennusteita laadittaessa huomioida nämä epälineaarisuudet.

Yleisimmin tällä hetkellä käytössä olevat oppivat ja älykkäät mallinnustekniikat ovat *sumea logiikka* ja *neurolaskenta*. Sumeiden järjestelmien toiminta perustuu epätasämallisyyden ja epävarmuuden tarkasteluun. Epätasämallisyyttä on pyritty mallintamaan tarkoitukseen erityisesti formuloidun logiikan ja joukko-opin avulla. Epävarmuutta puolestaan käsitellään perinteisen todennäköisyyslaskennan avulla. Sumean logiikan järjestelmät muistuttavat läheisesti *asiantuntijajärjestelmiä* ja ne soveltuvat parhaiten erilaisten säätöjärjestelmien mallintamiseen. Neurolaskenta puolestaan soveltuu erityisesti epälineaaristen järjestelmien mallintamiseen ja suuresta informaatiomäärästä vaikeasti havaittavien riippuvuuksien selvittämiseen. Sen avulla pystytään ennustamaan myös ennenkokemattomia tilanteita. Johtuen tässä diplomityössä mallinnettavien järjestelmien epälineaarisuudesta ja riippuvuuksien monimutkaisuudesta, valittiin käytettäväksi mallinnustekniikaksi neurolaskenta.

Neuroverkkomallinnuksessa ei edes yritetä rakentaa kaiken kattavaa matemaattista mallia, vaan koko järjestelmää käsitellään ikäänkuin mustana laatikkona. Mallintaminen tapahtuu järjestelmää kuvaavien syöte-vaste-parien avulla opettamalla. Mallien toimivuuden kannalta on tärkeää, että suunnitteluvaiheessa pystytään oleellinen järjestelmää kuvaava informaatio erottamaan merkityksettömästä. Tästä syystä diplomityön kirjallisuusosassa käydään suhteellisen tarkasti läpi hajautetun painatusjärjestelmän osakomponenttien ominaisuuksia. Tavoitteena on neurolaskennan avulla kehittää insinöörityökaluja, joita voidaan hyödyntää hajautetun painatusjärjestelmän suorituskyvyn ennustamisessa. Näitä työkaluja voitaisiin edelleen hyödyntää esimerkiksi hajautettujen tulostusjärjestelmien suunnittelussa ja työnohjauksessa.

Diplomityön kirjallisuusosassa käydään läpi elektronisten painatusjärjestelmien ominaisuuksia ja teknistä toteutusta. Lähdemateriaalina on käytetty kirjoja, artikkeleita, raportteja, sekä sarja- ja konferenssijulkaisuja. Erityisesti on paneuduttu tiedonsiirto-tekniikoiden, tiedosto- ja tulostinpalvelimien, rasterointiprosessoreiden, sekä itse tulostimien ominaisuuksiin. Lisäksi on selvitetty neurolaskennan matemaattista taustaa ja tapoja hyödyntää kyseistä tekniikkaa mallinnuksessa. Tulostettavan materiaalin generointiin käytettyjä tekniikoita ja sovelluksia ei tässä diplomityössä käsitellä lainkaan. Malleja kehitettäessä tehtiin oletus, että tulostettava informaatio on valmiiksi Postscript-koodattua eli tiedostoja käsitellään ainoastaan tulostusjärjestelmään syötettävänä raaka-aineena.

Kokeellisen osan ensimmäisessä luvussa esitellään laadittu koesuunnitelma, jossa käydään läpi muun muassa koeympäristön tekniset ominaisuudet ja kuvataan suoritettavat kokeet. Tämän jälkeen paneudutaan neuroverkkomallien kehittämiseen liittyvään problematiikkaan. Neuroverkkomallia rakennettaessa ensimmäinen tehtävä oli tulostetun opetusmateriaalin analysointi järjestelmän suorituskyvyn merkittävästi vaikuttavien materiaaliominaisuuksien selvittämiseksi. Seuraavaksi esitetään perustelut neuroverkkomallin kehitysvaiheessa tehdyille algoritmivalinnoille ja sen arkkitehtuuria koskeville ratkaisuille. Itse mallintaminen päädyttiin jakamaan kahteen osaan. Ensimmäisessä vaiheessa pyritään neuroverkoilla ennustamaan yksittäisten tulostimien suorituskykyä materiaalista selvitettyjen parametrien avulla. Neuroverkkoja rakennettiin useampia sen selvittämiseksi mille tasolle materiaalin kuvaus on viетävä, jotta ennusteet olisivat tarkkoja. Samassa yhteydessä rakennettiin myös yleinen tulostinmalli järjestelmään liitettävälle tulostimille. Sen syöteinä käytetään materiaaliominaisuuksien lisäksi tulostimien ominaisuuksia. Toisessa vaiheessa näiden tulostinkohtaisten ennusteiden avulla pyritään ennustamaan koko hajautetun järjestelmän suorituskykyä. Mallin toimivuuden varmistamiseksi muodostettiin testiympäristössä useita tulostusarkkitehtuureja. Lopuksi käydään läpi mahdollisia jatkokehitysmahdollisuuksia, sekä esitetään johtopäätökset ja yhteenvedo. Liitteissä on esitetty käytetyn opetus- ja testiaineiston rakenne, sekä testitulokset, niiltä osin kuin niitä ei ole liitetty osaksi varsinaista tekstiä.



## 2

# HAJAUTETUT PAINATUSJÄRJESTELMÄT

## 2.1 Määritelmä

Termiä *hajautettu painaminen* käytetään tämän diplomityön yhteydessä kuvaamaan konventionaalisiin painomenetelmiin rinnastettavaa tulostamista<sup>1</sup>, joka tapahtuu tietokoneverkon kautta mahdollisesti useammalle tulostimelle samanaikaisesti. Varsinaisesta tekniikasta yleisemmin käytetty termi on *elektroninen painaminen*<sup>2</sup>, jolla viitataan menetelmiin joiden avulla on mahdollista suorittaa tulostus suoraan tietokoneelta siten, että jokaisen tulostettavan sivun sisältö on muista sivuista poikkeava. Termien erona voidaan pitää, että puhuttaessa *hajautetusta painamisesta* tapahtuu tulostaminen fyysisesti eri paikassa kuin informaation käsittely ja varastointi. Itse asiassa tulostin/tulostimet voivat teoriassa sijaita vaikka maapallon toisella puolella, ja niiden lukumäärä ei ole millään tavoin rajoitettu. Tämä ei tietenkään sulje pois mahdollisuutta kutsua tämän tyyppistä tulostamista *elektroniseksi painamiseksi*, mutta kääntäen tämä ei päde. Suoraan tietokoneelta siihen kytketyllä tulostimella tulostamista ei voida tässä diplomityössä käytetyn määritelmän perusteella kutsua *hajautetuksi painamiseksi*.

Painatuksen hajauttaminen voidaan toteuttaa eriasteisesti. Periaatteessa voidaan erottaa kolme erilaista toimintamallia: alueellisesti hajautettu painaminen, vähittäisjakeluun hajautettu painaminen ja loppukäyttäjälle hajautettu painaminen. Alueellisesti hajautetulla painatuksella tarkoitetaan lähinnä erilaisten satelliittipainojen käyttämistä. Tämän tyyppiset elektronista painatusta hyödyntävät ratkaisut rakentuvat yleensä suurtehotulostimien ympärille, eli periaatteessa kysymys on edelleen keskitetystä tulostamisesta. Vähittäisjakeluun soveltuvissa järjestelmissä voidaan tuotteista riippuen hyödyntää jo huomattavasti alhaisemman teholuokan tulostimia. Loppukäyttäjille hajautettu painaminen on mallintamisen kannalta vaativin, mutta myös mielenkiintoisin, koska tulostukseen voidaan käyttää periaatteessa hyvin erityyppisiä tulostimia. /30/

Painotuotannon hajauttaminen tiedonsiirtoverkkojen kautta asettaa korkeita vaatimuksia käytetyille datasiirtoyhteyksille ja tulostimille. Tällä hetkellä suurin ongelma maantieteellisesti laajalle alueelle hajautetuissa painatusjärjestelmissä on käytetyt siirtoyhteydet. Perinteisten painotuotteiden datamäärät liikkuvat parhaimmillaan gigabiteissa, jolloin esimerkiksi valinnaiset 64 kbit/s ISDN(Integrated Services Digital Network)-yhteydet ovat aivan liian hitaita tällaisten datamäärien siirtämiseen. Kiinteillä yhteyksillä päästään pidemmällä matkoilla riittäviin siirtonopeuksiin, mutta korkeat kustannukset ja riittämätön peitto muodostuvat tällöin ongelmaksi. Ratkaisuksi voi osoittautua uusi ATM(Asynchronous Transfer Mode)-tekniikka, jonka siirtonopeudet ovat luokkaa 155-625 Mbit/s. Tekniikan on ennustettu leviävän laajamittaisesti yrityskäyttöön jo muutaman vuoden sisällä ja kotitalouksiin aikajänteellä 10-20 vuotta. /29/ Lähi- ja alueverkkoympäristöissä,

1. Termejä *painaminen* ja *tulostaminen* käytetään toistensa synonyymeinä.

2. Myös *tietokonepainaminen*.



joissa datasiirtonopeudet ovat suuremmat kuin suuralueverkoissa, pullonkaulaksi muodostuu yleensä rasterointiprosessorin teho. Tässä diplomityössä datasiirtonopeus ei muodostunut ongelmaksi, koska testitiedostot olivat kohtuullisen kokoisia (<20 Mt) ja lisäksi testit suoritettiin lähiverkkoympäristössä, jonka siirtokapasiteettia voidaan teoriassa pitää tämän tyyppisille tiedostoille riittävänä.

Tulostuksen hajauttamisessa tietokoneverkon kautta on monia hyötynäkökohtia verrattuna suuren kapasiteetin omaavaan keskitettyyn tulostamiseen:

1. Yhden kalliin suurtehotulostimen sijasta voidaan hankkia useita halvempia tulostimia kapasiteetin siinä kärsimättä.
2. Suurella joukolla käyttäjiä on vaivattomasti mahdollisuus hyödyntää järjestelmään verkon kautta liitettuja tulostimia.
3. Hajautetuissa järjestelmissä tulosteet ovat useimmiten saatavissa hyvin paljon lähempää kuin keskitetyissä tulostusjärjestelmissä.
4. Tulostettavan materiaalin siirtäminen tulostimelle on varsinkin lähiverkoissa vaivatonta ja suhteellisen nopeaa, mitään fyysisiä siirtovälineitä, kuten korppuja, magneettinauhoja tai siirtokovalevyjä ei tarvita.
5. Jos käytössä on useita tulostimia voidaan laajempien töiden osia tietyin edellytyksin tulostaa samanaikaisesti.
6. Järjestelmän työnohjausta kehittämällä kuormitusta pystytään jakamaan tasaisemmin.

Tällä hetkellä hajautetun painatusjärjestelmän kannalta merkittäviä ovat erityisesti kohdat 1), 2), 3) ja 4). Kohta 5) on vielä kysymysmerkki, koska ei ole pystytty rakentamaan järjestelmää, jossa tulostettavan materiaalin jakaminen osiin ja näiden osien käsittely erillisinä kokonaisuuksina olisi mahdollista. Sellaisissa tapauksissa ongelmaa ei tietenkään ole, jossa materiaali on jo valmiiksi osissa. Tähän liittyvään problematiikkaan palataan tarkemmin kappaleessa 2.3. Kohta 6) sivuaa hyvin pitkälle tämän diplomityön aihealuetta. Tehokkaan työnohjauksen edellytyksenä on, että hajautetun painatusjärjestelmän suorituskyky pystytään ennustamaan riittävällä tarkkuudella. Tällä hetkellä suurimpina ongelmoina/puutteina hajautettujen ja elektronisten painojärjestelmien kohdalla voidaan pitää värien puuttumista, tulostuksen rajallista laatua ja on-line-jälkikäsittelyn alkeellisuutta/puuttumista. Myös mahdolliset tuoteformaattivaihtoehdot ovat hyvin suppeat, jos ylimääräisiä jälkikäsittelyvaihtoehtoja halutaan välttää.

Johtuen *elektronisen painatuksen* luonteesta, ei perinteisiä fotomekaanisia "impact"-painomenetelmiä kuten kohopainoa, laakapainoa ja syväpainoa voida kyseisissä järjestelmissä hyödyntää. Näissä menetelmissä painotyön jokaista sivua varten valmistetaan kertakäyttöinen painolevy, jonka sisältämä informaatio ei painoprosessin aikana ole millään tavoin muunneltavissa. Ainoa mahdollisuus on kokonaan uuden levyn valmistaminen. Tästä syystä elektronisissa painojärjestelmissä hyödynnetään pääasiassa NIP- eli "non-impact printing"-menetelmiä, joissa painolevyn vastine on yleensä jokin kestävä käyttöinen reseptori, jolle latentti kuva generoidaan uudelleen jokaisella pyörähdyskerralla. Tämä mahdollistaa elektronisille painojärjestelmille tunnusomaisen sivun sisältöinformaation vaihtumisen tulostettavalta sivulta toiselle. NIP-tekniikoihin palataan tarkemmin kappaleessa 3.5, jossa käydään läpi elektronisiin painojärjestelmiin soveltuvien tulostimien ominaisuuksia.

Tällä hetkellä elektroninen painaminen pystyy kilpailemaan konventionaalisten menetelmien kanssa painoksien ollessa pieniä. Tämä johtuu lähinnä NIP-tekniikoiden asettamista rajoituksista tulostusnopeudelle. Lisäksi on huomioitava tulostuksen laatu, eli yleisimmin elektronisissa painatusjärjestelmissä käytetty elektrofotografiatekniikka ei vielä tällä hetkellä pysty laadullisesti kilpailemaan perinteisten painomenetelmien kanssa, kun kyseessä ovat korkealaatuiset painotyöt. Tällä hetkellä elektronisesti painetaan suuremmassa mitta-kaavassa pääasiassa oppimateriaalia, käyttöohjeita ja muuta usein päivitettävää, runsaasti vaihtuvaa informaatiota sisältävää materiaalia. Oletettavaa kuitenkin on, että lähitulevaisuudessa tekniikkaa ryhdytään hyödyntämään myös esimerkiksi normaalissa kirjatut-



nossa. Erityisesti tämä tuntuu houkuttelevalta sellaisten kirjojen tapauksessa, joiden painokset ovat pieniä ja kysyntä vaihtelevaa. /2,9,34,35,39/

Elektronista/hajautettua painamista voidaan pitää tulevaisuuden nousevana teknologiana siitä syystä, että koko mediakenttä on kokonaisuutena kehittymässä aivan uuden tyyppiseksi. Voidaan sanoa, että tänä päivänä on käynnissä informaation välitystapojen kokonaisvaltainen muuttuminen. Viestintäprosesseissa ollaan siirtymässä erätuotannosta on-demand tuotantoon. Tällä tarkoitetaan, että varastoitu informaatio tuotteistetaan vasta siinä vaiheessa, kun sille on kysyntää. Myös teknologioita ja järjestelmiä pyritään integroidaan mahdollisimman pitkälle eli off-line ympäristöistä ollaan vähitellen siirtymässä on-line ympäristöihin. Muutos alkaa vaikuttaa myös itse tuotteisiin eli niitä ollaan kehittämässä ei-interaktiivisista yhä enemmän interaktiivisuuden suuntaan. Käyttäjälle halutaan tarjota mahdollisuus henkilökohtaisesti rakentaa ja profiloida tuotteet. /36/

## 2.2 Kapasiteettiin vaikuttavat tekijät

### 2.2.1 Hajautetun tulostusjärjestelmän komponentit

Aiemmissa tutkimuksissa on selvästi havaittu, että tulostusjärjestelmän osakomponenttien ominaisuudet vaikuttavat hyvin selvästi muodostuvaan suorituskykyyn. Hajautetun painatusjärjestelmän voidaan teoriassa katsoa muodostuvan seuraavista osista (kuva 1):

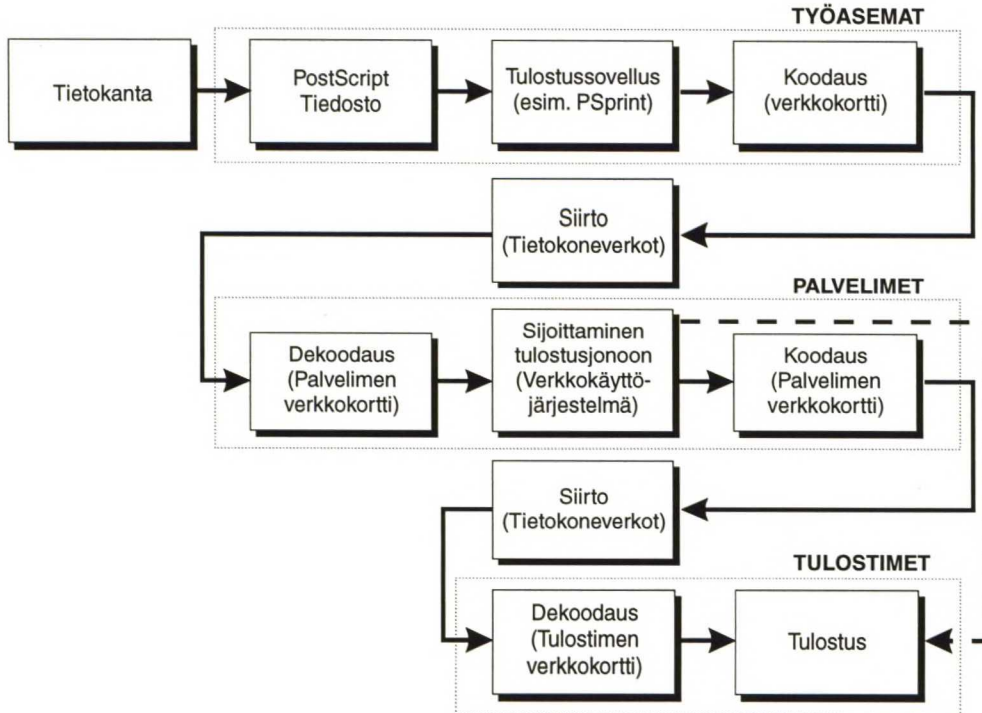
- Työasema(t).
- Tietokoneverkko/verkot.
- Palvelin/palvelimet.
- Tulostin/tulostimet.

Suurin osa tietokantojen ja työasemien ominaisuuksista on tässä työssä sivuutettu, koska materiaalin käsittelyyn ja sivunkuvauksen muodostamiseen kuluneen ajan vaikutusta suorituskykyyn ei tässä työssä huomioida lainkaan. Tämän tyyppisessä järjestelmässä kokonaistulostusaika muodostuu tiedoston siirtoon, sen väliaikaiseen tallentamiseen (tulostusjonot), tulkitsemiseen (rasterointiprosessori) ja varsinaiseen tulostamiseen kuluva ajasta. Koska tiedoston siirtoon kuluva aika vaikuttaa kokonaistulostusaikaan, voidaan ainoana suorituskyvyn kannalta merkittävänä työaseman ominaisuutena pitää tapaa, jolla kytketyminen tietokoneverkkoon on toteutettu. Käytännössä tämä tarkoittaa verkkokortin nopeutta. On selvää, että rakenteellisten dokumenttitietokantojen yleistyessä, myös sivunkuvauksen muodostaminen muuttuu on-line-toiminnoiksi. Tällöin työaseman muut ominaisuudet muuttuvat painatusjärjestelmän suorituskyvyn kannalta merkittäviksi tekijöiksi.

Kuvaajassa esitetty "siirto"-laatikko voi sisältää useampia lähiverkkoja. Näiden yhdistämiseen voidaan käyttää esimerkiksi siltoja, reitittämiä tai alueverkkoja, joiden peittoalue voi olla muutamasta korttelista koko kaupunkiin. Jos yhdistettävät verkot sijaitsevat fyysisesti vielä laajemmalla alueella, saatetaan niiden väliseen tiedonsiirtoon tarvita myös suuralueverkkoja. Koska erilaiset tiedonsiirtoverkot näyttävät merkittävää osaa hajautetuissa painatusjärjestelmissä, on niitä ja niiden ominaisuuksia käsitelty tässä työssä suhteellisen laajasti luvussa 3.

Koska järjestelmä voi sisältää useampia lähiverkkoja, myös palvelimia tarvitaan tällaisissa tapauksissa useampia ja tietokoneverkoissa voi myös lokaalisti olla useampia palvelimia, jolloin niillä jokaisella on yleensä oma tehtävänsä. Yleisin jako lienee tiedosto- ja tulostinpalveluiden erottaminen. Erillinen tulostinpalvelin ei ole välttämättömyys, koska myös tiedostopalvelin pystyy tarjoamaan vastaavat palvelut. Syy siihen, miksi tehtäviä jaetaan

erillisille palvelimille on yksinkertainen: näin saadaan jaettua kuormitusta tasaisemmin ja tällä on yleensä merkittävä vaikutus tietokoneverkon tehokkuuteen. Tulostimille asetettavat vaatimukset riippuvat hyvin pitkälle elektronisen painatusjärjestelmän käyttötarkoituksesta. Kuten jo aiemmin mainittiin, yleisimmin elektronisissa tulostusjärjestelmissä hyödynnetään NIP-tekniikoita ja erityisesti elektrofotografiaa. Mainittuihin tekniikoihin palataan tarkemmin luvussa 3, jossa käsitellään hajautetun painatusjärjestelmän teknistä toteuttamista.



**Kuva 1.** Yleinen malli tietokoneverkon ympärille rakennetusta hajautetusta painatusjärjestelmästä. Kiinteä viiva (—) kuvaa tilannetta, jossa kaikki järjestelmän osat ovat aidosti verkkoon kytkettyjä. Katkoviiva (---) kuvaa tilannetta, jossa tulostin/tulostimet on kytketty suoraan tiedosto/tulostinpalvelimeen.

## 2.2.2 Tulostettavan materiaalin rakenne ja sisältö

Ryhdyttäessä pohtimaan dokumenttien sisällön ja rakenteen vaikutusta hajautetun painatusjärjestelmän kokonaissuorituskykyyn, vaikuttavia tekijöitä on käytännössä kaksi. Ensinnäkin dokumentin sisältö vaikuttaa sen tiedoston kokoon, johon sivunkuvaus tallennetaan ja täten myös tiedoston siirtämiseen kuluvaan aikaan<sup>1</sup>. Toiseksi, mitä monimutkaisempia elementtejä dokumentti sisältää, sitä monimutkaisempi on myös siitä muodostettu sivunkuvaus. Sivunkuvaus vaikuttaa tulostusprosessin tehokkuuteen siinä vaiheessa, kun sitä ryhdytään tulkitsemaan ja kääntämään tulostimen ohjausinformaatioksi. Tämä johtuu siitä, että huolimatta tietotekniikan nopeasta kehityksestä elektronisten tulostusjärjestelmien pullonkaula on edelleen yleensä rasterointiprosessori. Mitä monimutkaisempia elementtejä dokumentti sisältää, sitä kauemmin kestää myös rasterointi eli kysymys on yksinkertaisesti laskentatehosta. Tiedostojen sisältö voidaan yleensä jakaa karkeasti kolmeen kategoriaan, jotka ovat

1. Siirtoaikaan vaikuttavia tulostusjärjestelmän teknisiä ominaisuuksia käsitellään tarkemmin luvussa 3.



- teksti,
- viivakuvat ja
- sävykuvat.

Näistä tekstin käsittely kuormittaa rasterointiprosessoria kaikkein vähiten ja sävykuvien eniten. Viivakuvien prosessointiaika sijoittuu yleensä näiden kahden välimaastoon. Tulostettaessa pelkkää tekstiä päästään yleensä hyvin lähelle tulostimien teoreettista koneist nopeutta ja ainoaksi tulostusnopeutta merkittävästi hidastavaksi tekstidokumenttien ominaisuudeksi on osoittautunut yhdellä sivulla esiintyvien fonttien lukumäärä. Tämä pätee ainoastaan siinä tapauksessa, että fonttivaihdoksia on sivulla niin runsaasti, että kirjoittimen työmuisti ei enää riitä tarvittavien fonttien käsittelyyn. Muistin loppuminen saa aikaan tulostusaikojen moninkertaistumisen, joten piirrettä voidaan pitää suorituskyvyn kannalta merkittävänä. Sillä minkä tyyppisiä muutoksia fonteissa tapahtuu ei ole havaittu olevan merkittävää vaikutusta.

Yleisimpiä käytettyjä viivakuvaformaatteja ovat CGM (Computer Graphics Metafile), EPS (Encapsulated Postscript) ja WMF (Windows Metafile). Näistä on kerrottu tarkemmin taulukossa 1. Viivakuvien tapauksessa tulostusajan kannalta merkittävimmäksi tekijäksi on osoittautunut niiden sisältämien kuvaobjektien lukumäärä. Niiden lukumäärän selvittäminen kuvatiedostosta on kuitenkin erittäin vaikeaa ellei jopa mahdotonta. On kuitenkin havaittu, että viivakuvatiedoston koko korreloi riittävällä tarkkuudella kuvaobjektien lukumäärän kanssa, joten viivakuvien tulostusaikoja selittävänä tekijänä voidaan käyttää kuvatiedoston kokoa.

Taulukko 1. Yleisimmät viivakuvaformaatit ja lyhyt kuvaus niiden ominaisuuksista. /24/

Viivakuvaformaatti	Ominaisuudet
<b>CGM</b> Computer Graphics Metafile (ISO 8632-1991)	<ul style="list-style-type: none"><li>• Tiedosto muodostuu peräkkäisistä grafiikkakomennoista (metatiedosto).</li><li>• Sisältää kaikki perusgrafiikkarutiinit, kuten esim. komennot viivojen, ympyröiden, monikulmioiden ja ellipsien piirtämiseen.</li><li>• CGM:llä voidaan koodata myös sävykuvia<sup>a</sup>.</li><li>• Ainoa virallisesti standardoitu grafiikkaformaatti tällä hetkellä.</li><li>• Perustuu hyvin pitkälle GKS:ssä eli <i>Graphical Kernel System</i>issä määriteltyihin standardeihin grafiikkarutiineihin.</li></ul>
<b>EPS</b> Encapsulated Postscript	<ul style="list-style-type: none"><li>• kts. taulukko 2</li></ul>
<b>WMF</b> Windows Metafile	<ul style="list-style-type: none"><li>• Tiedosto muodostuu listasta Microsoft Windowsin grafiikkafunktiokutsuja.</li><li>• Soveltuu käytettäväksi PC(personal computer)-ympäristössä Microsoft Windows-yhteensopivissa sovelluksissa.</li><li>• Tiedostot toteutettu rakenteisesti. Kooltaan ne voivat olla huomattavasti pienempiä kuin vastaavat bittikartat johtuen korkeamman tason kuvausominaisuuksista.</li></ul>

a. Ei hyödynnetty tässä diplomityössä.

Kuten edellä viivakuvien tapauksessa myös sävykuvat voidaan luokitella niiden formaatin, sekä tiedoston koon perusteella. Yleisesti käytettyjä sävykuvaformaatteja ovat esimerkiksi BMP (Bitmap), EPS ja TIFF (Tagged Image File Format). Näiden sävykuvaformaattien ominaisuuksia on tarkemmin käsitelty taulukossa 3. Näiden kuvaformaattien välillä ei ole koe-tulostuksissa todettu merkittäviä eroja tulostusaikoja vertailtaessa, tulostusajan ja Postscript-tiedoston koon välillä näyttäisi vallitsevan lineaarinen riippuvuus. Kuvan koolla tai resoluutiolla ei myöskään ole aiemmissa tutkimuksissa havaittu olevan selvää tulostus-

aikoja vääristävää vaikutusta. Tämän voidaan katsoa johtuvan siitä, että kuvatiedoston koko on hyvin pitkälle riippuvainen kuvan fyysisestä koosta ja sen resoluutiosta.

Taulukko 2. Yleisimmät sävykuvaformaatit ja lyhyt kuvaus niiden ominaisuuksista. /24/

Sävykuvaformaatti	Kuvaus/ominaisuudet
BMP Bitmap	<ul style="list-style-type: none"><li>• Yleisin tapa koodata ja esittää sävykuvia.</li><li>• Kuva-alue jaetaan elementeiksi, joita kutsutaan pikseleiksi. Jokaisella pikselillä on erillinen, sen vaaleutta/tummuutta tai väriä kuvaava arvo.</li><li>• Hyödynnetään lähes kaikissa sävykuvaformaateissa.</li></ul>
EPS Encapsulated Postscript	<ul style="list-style-type: none"><li>• Kuvatyyppi joko vektori tai bittikartta.</li><li>• Perusmuotoista Postscript-koodia.</li><li>• Tiedosto on formaatiltaan editoitavaa ASCII-tekstiä.</li><li>• Tarkemmin <i>Postscript</i>ia on käsitelty tämän diplomityön kappaleessa 2.3.</li></ul>
TIFF Tagged Image File Format	<ul style="list-style-type: none"><li>• Kuvatyyppi on bittikartta.</li><li>• Tiedostoon voidaan liittää tietoja esim. käytettävästä skannerista, monitorista tai tulostimesta.</li><li>• TIFF-tiedosto voidaan jakaa hierarkisesti kolmeen kerrokseen. Ensimmäisen muodostaa tiedoston otsikko (header), toisen yksi tai useampi osoitehakemisto (IFD eli Image File Directory) ja kolmannen varsinainen kuvadata, johon osoitehakemiston osoitteet viittaavat.</li></ul>

Kun järjestelmän suorituskykyä halutaan ennustaa tulostettavien tiedostojen sisällön ja rakenteen avulla ongelmaksi muodostuu, että yhdisteltäessä tekstiä ja useita kuvaformaatteja samaan tiedostoon, ei tulostusaika enää korreloi tiedoston koon kanssa. Toisin sanoen tiedoston kokonaistulostusajan ennustaminen sen sisältämien elementtien erikseen mitattujen tulostusaikojen perusteella ei onnistu. Selittäviä tekijöitä on vaikea löytää, mutta todennäköisesti kyseessä on Postscript-kielen ja käytettyjen taitto-ohjelmien Postscript-ominaisuuksien yhteisvaikutuksista, jotka eivät ole lineaarisia. /33, 40/ Koska aiemmissa tutkimuksissa ja myös tässä työssä käytetään sivunkuvaukseen ainoastaan Postscript-kieltä, on kappaleessa 2.3 käyty lyhyesti läpi sen yleisempiä piirteitä ja ominaisuuksia.

## 2.3 Postscript-sivunkuvauskieli

Kuten edellä mainittiin, merkittävää osaa hajautetun painatusjärjestelmän suorituskyvyn muodostumisessa näyttelee käytettävä sivunkuvauskieli tai pikemminkin sen ominaisuudet ja tehokkuus. Mitä tehokkaammin sivun ulkoasu on koodattu, sitä nopeammin se pystytään rasterointiprosessorilla tulkitsemaan ja muuttamaan tulostimen ohjaussignaalksi. Tällä hetkellä yleisimmät käytetyt sivunkuvauskielet ovat Adoben Postscript, Hewlett-Packardin PCL(Printer Control Language) ja HPGL(Hewlett-Packard Graphics Language), sekä Xeroxin InterPress. Ylivoimaisesti yleisin näistä on Postscript, joka on saavuttanut standardin aseman maailmanlaajuisesti. Tässä diplomityössä muut sivunkuvauskielet rajattiin käsittelyn ulkopuolelle, eli kaikki testimateriaalina käytetty data on Postscript-koodattua. Tähän vaikutti edellä mainitun lisäksi, että kaikki Graafisen tekniikan laboratorion tulostimet ovat Postscript-tulostimia. Koska Postscript-kieli on aihealueena erittäin laaja, ei sen ominaisuuksien perinpohjainen esittely tämän diplomityön puitteissa ole mahdollista. Seuraavassa on ainoastaan käyty lyhyesti läpi Postscript-kielen perusteita ja ominaisuuksia, jotta olisi helpompi ymmärtää mitkä ovat sen käytön edut ja mitä ongelmia sen käyttöön liittyy.



Postscript on yksinkertainen käännettävä ohjelmointikieli, jonka voima piilee sen erittäin tehokkaissa grafiikan käsittelyominaisuuksissa. Sen ensisijainen tehtävä on kuvata sivulle sijoitetun tekstin, grafiikan ja kuvien ulkoasu siten, että niiden tulostaminen on mahdollista laitteistoriippumattomasti. Postscript-koodia voidaan näin ollen hyödyntää myös järjestelmien välisessä sivunkuvauksen siirrossa. Postscript sivunkuvauksia voidaan esittää tietyllä tulostimella tai muulla output-laitteella (esimerkiksi monitorilla) syöttämällä se tätä ohjaavalle kääntäjälle. Postscript-kääntäjä tulkitsee koodin eli käytännössä suorittaa tiedoston sisältämät komennot koodin määräämässä järjestyksessä, kuten minkä tahansa ohjelmointikielen kääntäjä. Erona voidaan pitää, että lopputuloksena ei saada ajettavaa ohjelmaa vaan kuva sivusta, joka voidaan tämän jälkeen esittää halutussa muodossa eli esimerkiksi tulostaa paperille tai näyttää monitorilla. Normaalisti Postscript-muotoisen sivunkuvauksen luomisesta huolehtii käytetty sovellus eli esimerkiksi tekstinkäsittely- tai taitto-ohjelma. Postscript koodia kirjoitetaan suoraan yleensä ainoastaan silloin kun halutaan hyödyntää ominaisuuksia, jotka eivät ole sovelluksen kautta käytettävissä.

Kuten jo aiemmin mainittiin on Postscript kuten mikä tahansa ohjelmointikieli. Se sisältää kaikki tavalliset datatyypit, kuten numerot, merkkijonot ja taulukot. Lisäksi sillä voidaan toteuttaa ehdollisia rakenteita, silmukoita ja aliohjelmiä. Postscript ohjelmat luodaan, lähetetään ja tulkitaan aina täysin samassa muodossa eli mitään käännettyä tai koodattua esitysmuotoa koodista ei ole olemassa. Kaikki koodin esittämiseen käytetyt merkit ovat visuaalisesti tunnistettavissa. Tämä merkkipohjainen esitysmuoto helpottaa koodin luomista, manipulointia ja myös ymmärtämistä. Koodissa hyväksyttävät merkit valitaan yleensä hyvin suppeasta merkistöstä ja juuri tämä piirre tekee Postscriptista laitteisto- ja järjestelmäriippumattoman.

Postscript tiedosto muodostuu tavallisimmin kahdesta osasta. Nämä ovat niin sanottu prologi ja varsinainen koodi. Prologi sisältää sovelluskohtaiset määritelmät, joita hyödynnetään varsinaisessa koodissa. Sovelluksen ohjelmoija rakentaa prologin manuaalisesti ja se lisätään jokaisen sovelluksella generoidun Postscript-tiedoston alkuun. Se sisältää määritykset, jotka sovittavat sovelluksen output-toiminnot Postscript-kielen ominaisuuksiin. Varsinaisen koodin sovellus generoi automaattisesti dokumentin sisällön perusteella. Koodi sisältää viittaukset sekä Postscript primitiiveihin, että prologissa tehtyihin määrittelyihin. Varsinainen koodi on yleensä erittäin siistiä ja selkeää verrattuna prologiin, jonka sisältämästä koodista pyritään tekemään mahdollisimman kompaktia ja tehokasta. /7/

Koodin osalta ongelmaksi muodostuu, että vaikka Postscript-kieli mahdollistavat jokaisen sivun erottamisen koodissa omaksi kokonaisuudekseen, useat sovellukset on rakennettu siten, että ne eivät hyödynnä tätä ominaisuutta. Tästä on seurauksena, että kerran muodostettua Postscript-tiedostoa on erittäin hankala, ellei jopa mahdoton jakaa pienempiin osiin, joita pystyttäisiin esimerkiksi käsittelemään rinnakkaisilla prosessoreilla. Ongelma johtuu siitä, että Postscript on ominaisuuksiltaan kuten muut ohjelmointikielet eli se mahdollistaa aliohjelmien ynnä muiden vastaavien käytön sivun ulkoasun koodauksessa. Yhdellä sivulla käytetyt operaattorit voivat vaikuttaa siihen, miten seuraavia sivuja käsitellään. Tästä syystä Postscript-tiedosto joudutaan aina tulkitsemaan kokonaisuutena.

Johtuen edellä kuvatulla tavalla generoidun Postscript-koodin jakamattomuudesta ja sitä kautta sopimattomuudesta rinnakkaiskäsitteelyyn on Adobe pyrkinyt vaikuttamaan ohjelmistotalojen tapaan kehittää Postscript-koodia generoivia sovelluksia. Se on muun muassa julkaissut niin sanottuja "dokumenttien rakenne sopimuksia" (DSC eli Document Structuring Conventions), joita noudattamalla helpotetaan koodin tulkintaa ja siten myös rasterointia. Rasterointiprosessori pystyy erottamaan sivut näiden sääntöjen mukaan generoidusta Postscript-koodista yksiselitteisesti. Ongelma ei kuitenkaan ole poistunut näiden sopimusten avulla, koska ne ovat ainoastaan suosituksia ja niitä noudatetaan/tulkitaan hyvin väljästi. /4,7/



## 3

# HAJAUTETUN PAINATUSJÄRJESTELMÄN TEKNINEN TOTEUTUS

## 3.1 Yleistä tiedonsiirtoverkoista

### 3.1.1 Peruskäsitteitä

**Tiedonsiirrolla** tarkoitetaan signaalien siirtämistä tietystä paikasta yhteen tai useampaan tiettyyn paikkaan. Jos kyseessä on tiedonsiirto, jossa päätelaitteina käytetään tietokoneita tai niiden oheislaitteita, puhutaan **datasiirrosta**. /8/ Tiedonsiirto tapahtuu aina **lähettimen** ja **vastaanottimen** välillä tietyn **siirtomedian** kautta. Siirtomediat voidaan luokitella ohjattuihin ja ohjaamattomiin; molemmissa tapauksissa kommunikointi tapahtuu elektromagneettisten aaltojen avulla. **Ohjattujen medioiden** tapauksessa siirto tapahtuu fyysisesti rajattua kanavaa pitkin, joka voi olla esimerkiksi kierretty parikaapeli. **Ohjaamattomat mediat** puolestaan tarjoavat keinon lähettää elektromagneettisia aaltoja, mutta niitä ei pystytä lainkaan ohjaamaan. Esimerkkeinä voitaisiin mainita lähetykset ilmassa, tyhjiössä tai vedessä. Tässä työssä käsitellään ainoastaan datasiirtojärjestelmiä jotka hyödyntävät ohjattuja siirtomedioita.

Signaaleja, joita siirtomedian kautta välitetään on olemassa periaatteessa kahdenlaisia: analogisia ja digitaalisia. **Analoginen signaali** on jatkuva elektromagneettinen aalto, jota voidaan moduloida siirrettävällä informaatiolla. Modulointimenetelmiä on useita, joista mainittakoon yleisimmät eli taajuus- ja amplitudimodulointi. **Digitaalinen signaali** puolestaan muodostuu jännitepulsseista, joita muodostetaan niin ikään lähetettävän informaatian ohjaamana. Lähetettävä data on binäärimuotoista eli pulssilla voi olla ainoastaan kaksi arvoa, joko 0 tai 1. Nollaa voidaan kuvata esimerkiksi vakiollisella negatiivisella jännitteellä ja ykköstä vakiollisella positiivisella jännitteellä. Analogista ja digitaalista dataa voidaan lähettää molemmilla signaalityypeillä eli digitaalinen data ei välttämättä vaadi digitaalista signaalia. Tällaisissa tapauksissa kommunikoivat laitteet huolehtivat tarvittavista muunnoksista.

Nopea datasiirto asettaa kahden laitteen väliselle kommunikoinnille suuria vaatimuksia. Koska dataa siirretään yleensä bitti kerrallaan, tulee näiden bittien ajoituksen olla sama lähettäjällä ja vastaanottajalla. Tämän ajoittaminen voidaan toteuttaa kahdella tavalla eli asynkronisesti tai synkronisesti. **Asynkronisessa siirrosta** lähetettävä data pilkotaan 5-8 bitin palasiksi, joista jokainen edustaa yhtä merkkiä. Merkit erotetaan toisistaan aloitus ja lopetus biteillä. Tällä tavoin ikään kuin mahdollistetaan kommunikoivien laitteiden synkronointi jokaisen lähetetyn merkin jälkeen. **Synkronisessa siirrosta** puolestaan lähetetään jatkuvaa bittivirtaa, joka sisältää ainoastaan lähetysten aloitus- ja lopetusmerkit. Jotta vastaanotin pystyisi erottamaan lähetetyt bitit, täytyy sen kellon olla synkronoitu lähettimen kellon kanssa. Tästä johtuu nimitys synkroninen siirto.



**Suoralla yhteydellä** tarkoitetaan yhteyttä joka on muodostettu kahden pisteen välille siten, että signaali kulkee suoraan lähettäjältä vastaanottajalle ilman mitään välikäsiä lukuunottamatta vahvistimia tai toistimia, joiden avulla signaalia voidaan voimistaa sitä kuitenkaan mitenkään muuttamatta. **Point-to-point yhteys** muodostuu kahden aseman välille silloin kun niiden välillä on suora yhteys, eikä mikään muu laite jaa samaa siirtomediaa. **Multipoint yhteyksissä** useampi kuin yksi laite hyödyntää samaa siirtomediaa. Yhteyksiä voidaan rakentaa kolmella tavalla. On olemassa simplex, half-duplex ja full-duplex ratkaisuja. **Simplex** yhteyksien tapauksessa liikenne tapahtuu ainoastaan yhteen suuntaan eli toinen laitteista on aina lähettäjä ja toinen vastaanottaja. **Half-duplex** yhteydessä molemmat laitteet voivat toimia sekä lähettäjänä, että vastaanottajana, mutta liikenne tapahtuu edelleen ainoastaan yhteen suuntaan kerrallaan eli asemat vuorottelevat lähettäjänä/vastaanottajana. **Full-duplex** yhteydessä molemmat asemat voivat lähettää ja vastaanottaa samanaikaisesti eli siirtomedia kuljettaa signaaleja kumpaankin suuntaan samanaikaisesti.

Nopeiden siirtoyhteyksien toteuttaminen on erittäin kallista ja koska kaksi kommunikoivaa asemaa harvoin käyttää median koko kapasiteettia, on kehitetty tekniikoita joiden avulla sitä pystytään jakamaan. Saman siirtomedian kapasiteetin jakamista useammalle samanaikaiselle käyttäjälle kutsutaan **multipleksoinniksi**. Multipleksointi voidaan toteuttaa kahdella perustekniikalla, jotka ovat taajuusjakoinen multipleksointi ja aikajakoinen multipleksointi. **Taajuusjakoisessa multipleksoinnissa** jaetaan siirtomedian kaistanleveyttä useammalle käyttäjälle. Tämä on mahdollista ainoastaan siinä tapauksessa, että median kaistanleveys on suurempi kuin lähetettävien signaalien vaatimat kaistanleveydet yhteensä. Taajuuskaistoihin jako tapahtuu käyttämällä eritaajuisia kantoaaltoja. **Aikajakoinen multipleksointi** puolestaan toteutetaan jakamalla lähetettävät datavirrat sopiviksi palasiksi ja lomittamalla nämä samalle siirtomedialle. Edellytyksenä on, että käytettävän median datasiirtonopeus ylittää multipleksoitavien digitaalisten signaalien yhteenlasketun datasiirtonopeuden.

Tiedonsiirtoverkoissa käytetään sovelluksesta riippuen kolmea kanavajakotekniikkaa. Tällä tarkoitetaan tapaa, jolla signaali hyödyntää siirtomediaa. Käytössä olevat tekniikat ovat kantataajuinen(Baseband)-, laajakaistainen(Broadband)- ja kantoaalto-siirto. **Baseband** järjestelmissä hyödynnetään aina digitaalisia signaaleja ja niin sanottua johtokoodausta (esimerkiksi Manchester). Baseband-järjestelmissä signaali varaa aina koko kaistanleveyden käyttöönsä. **Broadband** systeemeissä signaalit ovat puolestaan aina analogisia ja tämä mahdollistaa multipleksoinnin eli median jakamisen taajuuskaistoihin. Baseband-järjestelmissä päästään maksimissaan muutaman kilometrin siirtomatkoihin, kun taas Broadband-järjestelmissä etäisyydet voivat olla kymmeniä kilometrejä. **Kantoaalto-lähetyksissä** kantoaaltona toimivaa signaalia moduloidaan lähetettävällä bittivirralla. /19,41,42/

### 3.1.2 Siirtomediat

Siirtomedioiden **kanavakapasiteettia** määritettäessä on kiinnitettävä huomiota neljään tekijään:

- **Siirtonopeus:** Tällä tarkoitetaan nopeutta, jolla dataa voidaan siirtää ja sen yksikkö on useimmiten bittia sekunnissa (bit/s).
- **Kaistanleveys:** Kaistanleveys kertoo sen, kuinka paljon lähetin ja käytettävä siirtomedia rajoittavat lähetettävän signaalin taajuutta. Yksikkönä käytetään yleensä Hertziä (Hz).
- **Kohina:** Siirtokanavan aiheuttama keskimääräinen kohina.
- **Virhesuhde:** Kertoo virheellisten bittien osuuden. Binäärisiirrossa virhe muodostuu silloin, kun lähetetty nolla tulkitaan vastaanotettaessa ykköseksi tai ykkönen nollassi.



Kaikki nämä tekijät ovat kiinteästi riippuvaisia toisistaan. Lähtökohtana voidaan pitää, että mitä nopeammin dataa halutaan siirtää, sitä suuremmalla taajuudella se on lähetettävä ja sitä suurempi kaistanleveys vaaditaan siirtomediaalta. Tämä on pääasiassa kustannuskysymys eli mitä suurempi kaistanleveys halutaan käyttöön, sitä kalliimpaa on yhteyden toteuttaminen. Kaikilla käytetyillä medioilla on käytännössä rajallinen kaistanleveys, johtuen joko fysikaalisista tekijöistä tai tarkoituksellisesta rajoittamisesta esimerkiksi medioiden välisten häiriöiden minimoimiseksi. Tavoitteena on saavuttaa mahdollisimman suuri datasiirtonopeus pienimmällä mahdollisella virhesuhteella käytössä olevalla kaistanleveydellä. Rajoittavaksi tekijäksi muodostuu kohina eli käytetystä mediasta ja/tai ympäristöstä aiheutuva ei-toivottu, hallitsematon signaalin muuttuminen/sen laadun heikkeneminen.

Taulukossa 3 on käyty läpi tänä päivänä yleisimmin käytettyjen siirtomedioiden maksimi tiedonsiirtonopeudet ja kaistanleveydet, sekä pisin mahdollinen yhteysväli, joka voidaan toteuttaa signaalia välillä vahvistamatta. Huomattavaa on, että taulukossa esitetyt luvut ovat teoreettisia maksimiarvoja. Riippuen yhteyden toteutustavasta, siirtomedian laadusta ja sovelluksesta ne saattavat vaihdella huomattavasti. /41,42/

**Taulukko 3.** Yleisimmin käytetyt ohjatut mediatyypit ja niiden tärkeimmät ominaisuudet. Arvot ovat teoreettisia maksimiarvoja. /42/

Kaapelityypit	Tiedonsiirtonopeus	Kaistanleveys	Toistimien väli
Kierretty parikaapeli	4 Mbit/s	250 kHz	2-10 km
Koaksiaalikaapeli	500 Mbit/s	350 MHz	1-10 km
Optinen kuitu	2 Gbit/s	2 GHz	10-100 km

**Kierretty parikaapeli** muodostuu kahdesta eristetystä kuparijohtimesta, jotka on spiraalimaisesti kierretty toistensa ympärille niiden välisten elektromagneettisten häiriöiden minimoimiseksi. Parikaapeli soveltuu sekä analogiseen, että digitaaliseen tiedonsiirtoon ja se on vielä tällä hetkellä mainituista kolmesta ylivoimaisesti yleisin. Tämä johtuu siitä, että lähes koko puhelinverkko rakentuu parikaapelista ja näin ollen suurin osa esimerkiksi modeemien kautta hoidetusta tietoliikenteestä kulkee parikaapeleissa. Nykyään parikaapelia käytetään myös rakennusten sisäisten lähiverkkojen toteuttamiseen. Sen etu on halpa hinta ja asennuksen helppous, mutta sen kanavakapasiteetti ei riitä hoitamaan kovinkaan raskasta liikennettä ja lisäksi datansiirtonopeuden kasvaessa siirtoyhteyden maksimipituus lyhenee.

**Koaksiaalikaapeli** muodostuu kierretyn parin tavoin kahdesta johtimesta, mutta sen rakenne mahdollistaa huomattavasti laajemman taajuusalueen hyödyntämisen datasiirrosta. Kaapeli rakentuu sisemmästä ja ulommasta johtimesta. Sisempi johdin on erotettu ulommasta eristekerroksen avulla ja ulompana johtimena toimii eristekerrosta ympäröivä kuorikerros, joka edelleen on eristetty ympäristöstään. Koaksiaalikaapeli on tällä hetkellä yleisin lähiverkkoratkaisuissa käytetty kaapelityyppi. Sen hyviä puolia ovat suhteellisen edullinen hinta ja useimpiin sovelluksiin riittävä datasiirtonopeus.

**Optinen kuitu** on ohut (2-125 µm) siirtomedia, joka pystyy välittämään optisia säteitä. Kaapeli muodostuu kolmesta kerroksesta, jotka ovat ydin, sen päällyste ja vaippa. Ydin on optisen median sisin osa ja se muodostuu yhdestä tai useammasta muovista tai lasista valmistetusta säikeestä. Jokainen ytimen säie on päällystetty kerroksella muovia tai lasia, jonka optiset ominaisuudet poikkeavat ytimen optisista ominaisuuksista. Uloin kerros on vaippa, joka ympäröi yhtä tai useampaa päällystettyä säiettä. Se suojaaa niitä kosteudelta, murtumiselta ja kulumiselta. Optinen kuitu on ominaisuuksiltaan ylivoimainen verrattuna kierrettyyn pariin tai koaksiaalikaapeliin. Sen kaistanleveys on moninkertainen, se on



ohuempaa ja kevyempää, sen vaimennus on huomattavasti alhaisempi, se ei ole läheskään niin häiriöherkkää eli sähkömagneettiset kentät eivät vaikuta siihen ja se mahdollistaa huomattavasti pidemmät siirtomatkat ilman toistimia. Ainoa sen käyttöä rajoittava tekijä on hinta eli, datasiirtoyhteyksien toteuttaminen optisella kuidulla on huomattavasti kalliimpaa kuin kierretyllä parilla tai koaksiaaalikaapelilla. Optista kuitua hyödynnetään jo lähes kaiken tyyppisissä sovelluksissa suuralueverkoista lähiverkkoihin, aina kun erittäin nopea datasiirto on välttämättömyys /11,41,42/

### 3.1.3 Verkkojen luokittelu

Tiedonsiirtoverkot voidaan jakaa seuraaviin kategorioihin:

1. Kytkentäiset verkot
  - a. Piirikytkentäiset verkot.
  - b. Pakettikytkentäiset verkot.
2. Lähetysverkot
  - a. Lähiverkot (LAN eli Local Area Network).
  - b. Alueverkot (MAN eli Metropolitan Area Network).

**Piirikytkentäiset verkot** muodostuvat siirtoyhteyksistä ja niitä yhdistävistä niin sanotuista solmuista. Yhteen solmuun voidaan kytkeä useampia yhteyksiä. Erilaisten kytkentätekniikoiden avulla muodostetaan lähettäjän ja vastaanottajan välille verkon kautta kiinteä yhteys. Yhteyttä ylläpidetään niin kauan kuin se katsotaan toiminnan kannalta tarpeelliseksi, tämän jälkeen yhteys puretaan. Kaikki tällä tavoin toteutettuun verkkoon kytketyt laitteet pystyvät kommunikoimaan keskenään, mutta jokainen laite tarvitsee ainoastaan yhden I/O(input/output)-portin. Jos yhteydet olisi toteutettu niin sanotun täysin kytketyn verkon avulla, jokaisen aseman välille olisi muodostettava kiinteä yhteys. Tällöin N:n aseman kytkemiseen tarvitaan  $N(N-1)/2$  duplex-yhteyttä ja N-1 I/O-porttia jokaisella laitteella. Piirikytkentäisissä järjestelmissä on myös monia ongelmia. Jokaisesta verkon solmusta jonka kautta yhteys kulkee, joudutaan varaamaan yksi kanava pelkästään tämän käyttöön. Yhden solmun kautta kulkevien yhteyksien maksimimäärä on näin ollen käytännössä hyvin rajallinen. Lisäksi esimerkiksi tyypillisessä terminaalin ja keskuskoneen välisessä liikenteessä väylä on suurimman osan ajasta kuormittamaton, mutta koko ajan varattu. Piirikytkentäinen datasiirto edellyttää myös, että lähettäjä ja vastaanottaja toimivat synkronoidusti, eli lähetys- ja vastaanottonopeuden on oltava sama.

Ratkaisuna näihin ongelmiin on kehitetty **pakettikytkentäiset verkot**. Ne toimivat periaatteessa samalla tavalla kuin piirikytkentäiset, mutta erona on, että pakettikytkentäisessä verkossa yhteys ei ole fyysinen vaan looginen. Informaatio siirretään määrätyn kokoisina datapaketteina, joista jokainen sisältää määräaseman osoitteen. Jokainen verkon solmu lukee paketin muistiinsa, tarkastaa osoitteen ja reitittää sen perusteella paketin seuraavaan sopivaan solmuun. Tällä tavoin vältetään kaikki edellä esitetyt ongelmat. Tiettyä verkkosolmun kanavaa ei tarvitse varata tietylle yhteydelle eli verkko ja solmut kuormittuvat ainoastaan silloin, kun dataa todella siirretään. Lisäksi datan lähetysnopeus voi olla sen vastaanottonopeudesta poikkeava.

**Lähiverkot** ovat tietyllä suhteellisen suppealla maantieteellisellä alueella hyödynnettäviä datasiirtoverkkoja. Tällainen alue on yleensä yksi rakennus tai muutaman rakennuksen muodostama kokonaisuus. **Alueverkot** ovat laajakaistaisia ( $> 2$  Mbit/s) lähinnä datasiirtopalveluita tarjoavia julkisia verkkoja. Niiden peittoalue voi vaihdella muutamasta korttelista aina kokonaisiin kaupunkeihin. Ne ovat toteutustekniikoiltaan hyvin paljon lähiverkkojen kaltaisia, vaikka niiden datasiirtonopeudet voivat olla hieman alhaisempia johtuen pidemmistä siirtomatkista. Niitä käytetään usein lähiverkkojen yhdistämiseen yhdeksi suureksi kokonaisuudeksi, joka tarjoaa käyttäjilleen normaalin lähiverkon palvelut. Lähiverkot ja alueverkot luokitellaan lähetysverkoiksi, koska niiden kautta lähetetty informaatio välittyy kaikille verkkoon kytketyille laitteille eli signaalin vastaanottaja ei ole

yhteyden muodostushetkellä yksiselitteisesti määrättävissä samalla tavoin kuin piirikyt-kentäisten verkkojen tapauksessa. Molemmat hyödyntävät datasiirrosta pakettilähetystek-niikoita eli muodostetut yhteydet ovat loogisia. /11,19,25,42,45/

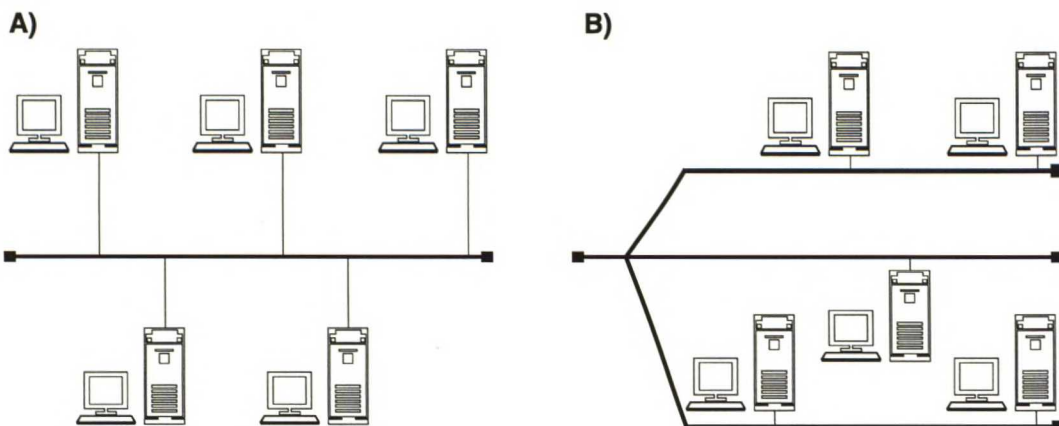
## 3.2 Lähiverkot (LAN) ja alueverkot (MAN)

Kun hajautettu painatusjärjestelmä toteutetaan verkkoympäristössä, on selvää että verkon rakenne ja ominaisuudet vaikuttavat sen suorituskykyyn ja kapasiteettiin. Seuraavassa on aluksi käyty läpi erilaisia tapoja toteuttaa verkon kaapelointi eli puhutaan verkkotopologi-oista. Sen jälkeen on paneuduttu lähiverkkotekniikoihin, jotka hyödyntävät näitä topologi-oita. Näistä on käsitelty yleisimmät eli CSMA/CD(Carrier Sense Multiple Access with Collision Detection), Token Ring, Token Bus ja FDDI(Fiber Distributed Data Interface). Lähiverkkotekniikat huolehtivat verkon tarjoamista alemman tason palveluista, jotka ovat varsinaiselle käyttäjälle suhteellisen näkymättömiä eli niitä hyödyntävät lähinnä käytettä-vät verkkosovellukset. Tällaisia palveluita ovat muun muassa lähetettävän datan koodaus/dekoodaus, lähetyksvuorojen jakaminen, liikenteen seuraaminen, törmäyksien havaitseminen jne. Käyttäjille suoraan tarjottavista palveluista huolehtivat yleensä verkko-käyttöjärjestelmät, joita on tarkemmin käsitelty kappaleessa 3.4, jossa käydään läpi verk-kopalvelimien ominaisuuksia.

### 3.2.1 Verkkotopologiat

#### Väylä/puutopologiat

Väylä/puutopologialla toteutetuille verkoille ominainen piirre on, että kaikilla verkkoon kyt-ketyillä työasemilla on yleensä yhtäaikaan pääsy verkkoon. Tästä huolimatta ainoastaan yksi asema kerrallaan voi hyödyntää verkkoa. Lähetys tapahtuu yleensä pakettina, joka sisältää vastaanottajan osoitteen. Kaikki verkkoon kytketyt asemat tarkkailevat verkossa liikkuvia paketteja, mutta kopioivat puskuriinsa ainoastaan ne, joissa on niiden oma osoite. Väylä- ja puutopologioilla ei ole toiminnan kannalta suurtakaan eroa. Väylätologia on ainoastaan yksinkertaistettu versio puutopologiasta eli se muodostuu ainoastaan yhdestä runkokaapelista. Puutopologiassa runkokaapelia on haaroitettu ja näitä haaroja kutsutaan oksiksi eli tästä johtuu nimi puutologia. Kuvassa 2 esitetty yksinkertainen malli molem-mista topologioista. /32,42,43/

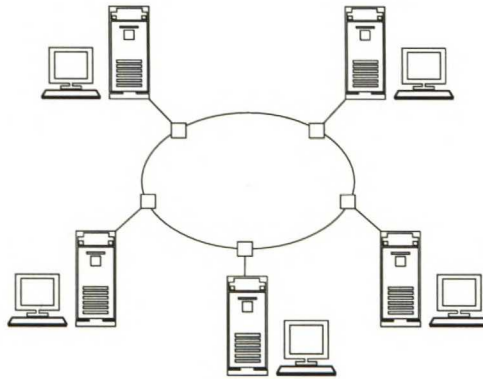


Kuva 2. a)Väylätologia. b) Puutologia. /43/



### Rengastopologia

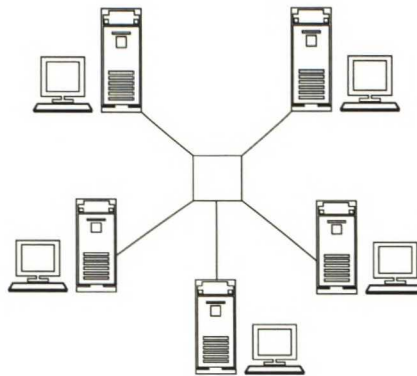
Rengastopologiassa verkko muodostuu sarjasta toistimia, jotka on kytketty toisiinsa point-to-point yhteyksillä siten, että ne muodostavat suljetun renkaan. Jokaista verkkoon kytkettävää laitetta varten tarvitaan oma toistimensa. Toistin on yksinkertainen laite, jolla on kaksi kytkentää verkkoon, ensimmäisen kautta se vastaanottaa dataa bitti kerrallaan ja toisen kautta data lähetetään samalla nopeudella eteenpäin: mitään puskurointia ei tarvita. Linkit ovat yksisuuntaisia eli datan kiertosuunta on verkossa aina sama. Data lähetetään pakettina, joka sisältää lähettäjän ja vastaanottajan osoitteet. Kun paketti kiertää verkossa, vastaanottava asema kopioi sen puskuriinsa ja lähettää sen jälkeen edelleen eteenpäin. Kun paketti palaa lähettäjälle, toimii se ikäänkuin kuittauksena ja se poistetaan verkosta. /32,42,43/



**Kuva 3.** Kuvassa on yksinkertainen esitys rengastopologian rakenteesta. Kuten havaitaan, kaapelointi muodostaa suljetun renkaan, johon työasemat on kytketty toistimien avulla. /43/

### Tähtitopologia

Tähtitopologiassa kaikki keskenään kommunikoivat työasemat on yhdistetty niin sanottuun keskuskyttimeen. Kuvassa 4 työasemien keskellä oleva laatikko kuvaa tätä kytkintä. Kun jokin työasemista haluaa lähettää dataa, pyytää se keskuskytkeä muodostamaan yhteyden lähettäjän ja vastaanottajan välille. Kun kytkin vastaanottaa yhteyspyynnön, muodostaa se piirikytkentäisen yhteyden asemien välille, tämän jälkeen se toimii tavalla kuin normaali point-to-point-yhteys. /32,42,43/



**Kuva 4.** Kaaviokuva tähtitopologian toteutuksesta. Keskellä kytkin, joka huolehtii yhteyksien muodostamisesta. /43/

Tähtitopologiaa voidaan soveltaa myös pakettikytkentäisissä lähetyksverkoissa. Tällöin asemat kytkeytyvät keskussolmuun kahdella point-to-point yhteydellä, joista toista käytetään vastaanottamiseen ja toista lähettämiseen. Kun jokin asema lähettää solmuun datapaketin, solmu välittää sen edelleen kaikille muille siihen kytketyille asemille. Kysymyksessä on siis topologia, joka on fyysinen tähti, mutta looginen väylä. /42/

### 3.2.2 Lähiverkkotekniikat

Edellä esitettyjä topologioita hyödynnettäessä, täytyy järjestelmän tarjota keinot verkon kautta tapahtuvan datasiirron hallintaan. Lähi- ja alueverkkoihin on käytännössä saatettu kytkeä tuhansia laitteita, joiden on jaettava käytössä oleva siirtokapasiteetti. Jotta kapasiteetin jakaminen olisi mahdollista, tulee pystyä kontrolloimaan näiden laitteiden pääsyä verkkoon. Ainoastaan tällä tavoin voidaan varmistaa kapasiteetin tehokas hyödyntäminen ja verkkokurin säilyminen. Menetelmiä kutsutaan yhteisellä nimellä MAC(Medium Access Control)-tekniikoiksi tai pelkästään verkkotekniikoiksi. Seuraavassa on lyhyesti käyty läpi yleisimmät tällä hetkellä lähi- ja alueverkoissa käytetyt MAC-tekniikat.

#### CSMA/CD (Ethernet)

CSMA/CD-tekniikka tunnetaan yleisemmin Ethernet<sup>1</sup> nimellä. CSMA/CD:n ominaisuudet on määritelty IEEE<sup>2</sup> 802.3 standardissa ja se on tällä hetkellä yleisin puu/väylä- ja tähtitopologioissa käytetyistä verkkotekniikoista. Tekniikka perustuu niin sanottuun kilpavarauksen menetelmään. Kun asema haluaa aloittaa lähettämisen, verkkokortti tarkastaa ensin, onko media vapaa vai onko jokin toinen asema jo lähettämässä dataa. Jos media on vapaa, aloitetaan datan lähettäminen. Siinä tapauksessa, että media on varattu verkkokortti odottaa niin kauan, että se vapautuu ja aloittaa lähettämisen välittömästi sen jälkeen. Koska kaikilla asemilla on samanaikaisesti pääsy verkkoon voi syntyä tilanne, jossa kaksi asemaa aloittaa lähettämisen lähes samanaikaisesti. Tällöin kumpikaan asema ei ole vielä havainnut toisen lähetystä ja tästä seuraa näiden lähetysten törmäys mediassa. Kun lähettävä verkkokortti havaitsee toisen signaalin, keskeyttää se oman lähetyksensä ja lähettää lyhyen törmäyksestä kertovan signaalin. Tämän signaalin tarkoituksena on varmistaa, että kaikki asemat havaitsevat törmäyksen. Törmäyksen havaitsemisen jälkeen kaikki lähettämistä yrittävät verkkokortit odottavat hetken aikaa ja yrittävät sen jälkeen aloittaa lähetysten uudelleen. Aika jonka jokainen verkkokortti odottaa valitaan satunnaisesti sen varmistamiseksi, että uutta törmäystä ei tapahtuisi.

CSMA/CD-tekniikkaa hyödyntävä verkko voidaan rakentaa joko kierretystä parikaapelista tai koaksiaalikaapelista. Standardi määrittelee viisi kantataajuiseen (Baseband) siirtoon soveltuvaa siirtomediastandardia. 10BASE2<sup>3</sup> ja 10BASE5 hyödyntävät koaksiaalikaapelia ja molemmat mahdollistavat 10 Mbit/s siirtonopeuden. Ne eroavat toisistaan segmentin maksimipituuden ja liittymien maksimimäärän suhteen. 10BASE2:ssa segmentin maksimipituus on 185 m ja laitteita voi olla kytkettynä korkeintaan 30. 10BASE5:ssä segmentin pituus voi maksimissaan olla 500 m ja siihen voidaan kytkeä korkeintaan 100 laitetta. Uusin tulokas on 100BaseT tekniikka, joka mahdollistaa 100 Mbit/s siirtonopeudet. Segmentin maksimipituus on suuresta siirtonopeudesta johtuen ainoastaan 100 m. 1BASE5:ssä ja 10BASE-T:ssä käytetään kierrettyä parikaapelia. 1BASE5:n maksimi siirtonopeus on 1 Mbit/s ja 10Base-T:n 10 Mbit/s. Ensimmäisen maksimi segmentin pituus on 250 m ja jälkimmäisen 100 m. Ainoa laajakaista(Broadband)-tekniikkaa hyödyntävä standardi on 10BROAD36, joka mahdollistaa 10 Mbit/s siirtonopeuden jopa 3,6 km pituisilla segmenteillä. /32,41,42/

---

1. Itse asiassa Ethernet on ainoastaan yksi kyseistä menetelmää hyödyntävistä verkkotekniikoista.  
2. The Institute of Electrical and Electronics Engineers  
3. Cheapernet.



## Token Ring

Token Ring (IEEE 802.5) on rengastopologioissa käytetty verkkotekniikka. Tekniikan suurin ero edelliseen on, että ainoastaan yhdellä asemalla kerrallaan on pääsy verkkoon. Tämä tarkoittaa, että jos jokin asema on lähettämässä dataa, muut eivät voi edes yrittää lähettämistä ja näin vältetään törmäykset. Verkkotekniikka perustuu verkossa kiertävään pieneen datapakettiin eli niin sanottuun *tokeniin*. Jos mikään renkaaseen kytketyistä asemista ei lähetä dataa, on *token* vapaa. Kun asema haluaa aloittaa lähettämisen, se odottaa niin kauan, että se havaitsee vapaan *tokenin* ja muuttaa sen varatuksi *tokeniksi*, koska verkossa ei nyt ole vapaata *tokenia*, mikään toinen asema ei voi aloittaa lähettämistä. Asema lähettää datan pakettina, joka sisältää sekä lähettävän, että vastaanottavan aseman osoitteen. Jokainen toistin lukee vuorollaan paketin, tarkastaa vastaanottajan osoitteen ja ellei se ole oma, välittää paketin verkossa eteenpäin. Tämä toistuu, kunnes paketti saapuu vastaanottavan aseman toistimelle, tämä kopioi sen aseman paikalliseen puskuriin, minkä jälkeen se lähettää paketin edelleen eteenpäin. Kun paketin lähettänyt asema saa sen takaisin, toimii se ikäänkuin kuittauksena, minkä jälkeen asema lähettää verkkoon vapaan *tokenin*. Seuraava lähetysvuoroa odottava asema havaitsee sen ja voi aloittaa lähettämisen. Token Ring standardi tukee ainoastaan kierretyllä parikaapelilla toteutettua kaapelointia ja se tarjoaa vaihtoehtoisesti kaksi tiedonsiirtonopeutta, joko 4 tai 16 Mbit/s. Yksi verkko voi sisältää maksimissaan 250 toistinta. /20,32,41,42/

Token Ring verkkojen katoamisesta on pitkään puhuttu ja vähitellen alkaa näyttää siltä, että näin on myös käymässä. Uusimmista tutkimuksista on käynyt ilmi, että Suomessa on puhtaita Token Ring ympäristöjä enää ainoastaan 8 %:lla yrityksistä, kun vastaava luku Ethernetin tapauksessa on 56 %. Lisäksi yritykset, joilla on käytössään sekä Token Ring-, että Ethernet-verkko, lähes poikkeuksetta ilmoittavat olevansa luopumassa Token Ringistä. Ainoa syy siihen, että Token Ring-ympäristöä edelleen ylläpidetään on lähes poikkeuksetta vanhojen laitteistojen asettamat vaatimukset. Ethernet on pääsemässä kilpailussa voitolle siitä syystä, että sen komponentit ovat huomattavasti halvempia ja sen laitteistotuki on suurempi. /22,38/

## Token Bus

Token Bus (IEEE 802.4)-tekniikan toimintaperiaate on täsmälleen sama kuin Token Ringin. Ainoa ero on, että Token Bus tekniikassa fyysiseen väylään kytketyt asemat muodostavat käytännössä loogisen renkaan. Tämä on toteutettu siten, että jokaisella asemalla on määrätty paikka järjestetyssä lähetysvuorolistassa. Listan viimeistä asemaa seuraa ensimmäinen eli näin muodostuu looginen rengas. Jokainen asema pystyy tunnistamaan verkosta sitä listassa edeltävän ja seuraavan aseman. Vastaavalla tavalla, kuin Token Ringin tapauksessa verkossa kiertää pieni *token*-paketti asemalta toiselle, lähetysvuorolistan osoittamassa järjestyksessä.

Standardi määrittelee kolme siirtomediavaihtoehtoa, jotka ovat Broadband<sup>1</sup>, Carrierband<sup>1</sup> ja Optical Fiber<sup>1</sup>. Kaksi ensimmäistä perustuvat koaksiaalikaapelointiin ja tukevat vaihtoehtoisesti joko 1:n, 5:n tai 10:n Mbit/s siirtonopeuksia. Ne hyödyntävät molemmat Broadband-lähetystekniikkaa. Ensimmäiselle ei ole määritelty maksimi segmentin pituutta ja toiselle se on noin 7,6 km. Kolmas tukee nimensä mukaisesti optiseen kuituun perustuvia järjestelmiä ja sen siirtonopeudet ovat joko 5, 10 tai 20 Mbit/s. Myöskään sen segmenteille ei ole määritelty maksimipituutta. /32,41,42/

## FDDI

FDDI on standardi verkkotekniikka erittäin nopeille rengasrakenteisille verkoille, jotka on toteutettu optisella kuidulla. FDDI tekniikassa hyödynnetään toimintaperiaatteeltaan samoja algoritmeja kuin Token Ringissä, mutta niitä on jouduttu kehittämään, jotta pystyt-

---

1. Standardinimiä vastaavalla tavalla kuin esimerkiksi 10BASE5, mistä syystä ei suomennettu.



täisiin hyödyntämään FDDI-protokollan mahdollistamaa 100 Mbit/s siirtonopeutta. Renkaaseen voi kuulua maksimissaan 1000 toistinta ja niiden välinen maksimietäisyys on 2,0 km. Tekniikkaa hyödynnetään usein MAN-verkkoja rakennettaessa. /37,41,42/

### 3.2.3 Lähiverkkojen yhdistäminen

Tietokoneverkkoympäristöissä muodostuu usein tarpeita ulottaa tiedonsiirtomahdollisuudet useisiin lähiverkkoihin. Yksi lähiverkko ei ulotu tarpeeksi laajalle alueelle tai järjestelmä muodostuu useista erillisistä lähiverkoista, jotka tulisi voida liittää yhteen. Seuraavassa on käyty läpi tekniikat, joiden avulla tämän tyyppinen yhdistäminen voidaan toteuttaa. Sitä, koska lähiverkoista muodostettu järjestelmä muuttuu alueverkoksi voidaan pitää lähinnä määrittelykysymyksenä, eikä sillä sinänsä ole käytännön merkitystä.

#### Toistin

Yksinkertaisin verkkojärjestelmien laajentamiseen sopiva laite on toistin. Itse asiassa se ei liitä yhteen useampia verkkoja vaan parantaa yksittäisen verkon ulottuvuutta. Toistin ainoastaan lukee verkosta saamansa viestin ja lähettää sen eteenpäin. Näin saadaan verkossa vaimentunut signaali vahvistettua ja siirron etäisyyttä huomattavasti jatketuksi. Signaalin vaimenemiseen esimerkiksi kaapelissa on monia syitä. Itse kaapelissa on sähköistä resistanssia, joka vaimentaa signaalia sen edetessä verkkoa pitkin. Vaimenemista aiheuttavat myös verkon liitokset ja kytkennät. Jos signaali vaimenee liiaksi, verkon asemat eivät enää pysty lukemaan sitä. Toistimen kannalta sen käsittelemän tiedon sisällöllä ei ole mitään merkitystä, sillä se vain uudistaa signaalin. Toistimia on kaksiporttisia ja moniporttisia. Kaksiporttisessa toistimessa toisesta portista vastaanotettu signaali lähetetään toiseen porttiin. Moniporttisessa toistimessa on useita portteja, esimerkiksi kahdeksan. Yhdestä portista vastaanotettu signaali toistetaan kaikkiin muihin portteihin. Toistin on teknisenä ratkaisuna yksinkertainen, luotettava ja halpa.

#### Silta

Sillan tehtävä on kahden lähiverkon fyysinen liittäminen toisiinsa. Käyttäjälle nämä sillalla yhdistetyt verkot näkyvät yhtenä ainoana verkkona eli kahdesta fyysisesti erillisestä verkosta muodostuu yksi looginen verkko. Silta asennetaan verkkoon kuten mikä tahansa muu asema ja se on useimmiten tavallinen mikrotietokone, johon on asennettu sovitinohjelma ja molempien yhteenliitettävien verkkojen kortit. Markkinoilta löytyy myös ainoastaan tähän tarkoitukseen rakennettuja laitteita eli niin sanottuja erillislaitteita. Siltojen avulla voidaan yhdistää myös eri verkkotekniikoita käyttäviä verkkoja eli esimerkiksi CSMA/CD- ja Token Ring-verkot. Käytännössä yhdistettävät verkot ovat yleensä samantyyppisiä, koska verkkotekniikoiden väliset lähetystapamuunnokset ovat erittäin monimutkaisia ja siten vaikeita toteuttaa yksinkertaisella sillalla. Silta hoitaa muiden toimintojensa ohella kaikki toistimen tehtävät. Siltoja käyttämällä voidaan parantaa myös lähiverkon tehokkuutta, koska sen kautta yhdistettyjen verkkojen sisäinen liikenne ei kuormita toista verkkoa.

#### Reititin

Reititin toimii periaatteessa samalla tavalla kuin silta, mutta kun yhdistettävien verkkojen määrä lisääntyy, myös mahdollisten reittivalintoja lukumäärä kasvaa, eikä silta enää pysty huolehtimaan sanomien ohjaamisesta. Koska yhdessä verkossa voi olla kiinni useampia reitittimiä on lähettävän aseman tiedettävä, minkä reitittimen kautta saadaan yhteys mihinkin asemaan tai verkkojärjestelmän osaan. Tästä syystä asemissa usein ylläpidetään niin sanottua reititystaulua, josta selviää jokaiseen verkon asemaan johtava lyhyin ja resursseiltaan paras reitti. Reitittimillä käytettyjä reititysalgoritmeja on olemassa useita. Yksinker-



taisimmat niistä ainoastaan lähettävät kaikki vastaanottamansa sanomat kaikkiin muihin kanaviin paitsi siihen, josta ne ottivat sanoman vastaan. Kehittyneemmissä algoritmeissa reitittimillä hyödynnetään niin sanottuja reititystauluja, joiden avulla löydetään reitti kohdeosoitteeseen. Reititin on huomattavasti kalliimpi ja monimutkaisempi laite kuin silta, mutta sen avulla voidaan hallita huomattavasti suurempia verkkokokonaisuuksia.

### Yhdyskäytävä (Gateway)

Edellä esitellyt ratkaisut on tarkoitettu avointen siirtojärjestelmien yhdistämiseen. Avointen järjestelmien käyttäminen on suositeltavaa, mutta jos joudutaan rakentamaan yhteyksiä suljettuihin järjestelmiin tarvitaan yhdyskäytäviä. Yhdyskäytävä on verkkojen väliseen liikenteeseen liittyvistä laitteista kaikkein monimutkaisin, mutta sen avulla voidaan yhdistää kaksi täysin toisistaan poikkeavaa verkkoympäristöä. Yhdyskäytävien ongelmaksi muodostuu niiden monimutkaisuus, minkä takia niiden hinta voi olla varsin korkea. Hintaa suurempi ongelma on yhdyskäytävien hitaus eli johtuen niiden tekemistä monimutkaisista muunnoksista, muodostuvat ne yleensä järjestelmien pullonkauloiksi. /11/

## 3.3 Suuralueverkot (WAN)

Edellä käsiteltiin lähiverkkoja, sekä niiden yhdistämiseen tietyllä rajatulla alueella käytettyjä tekniikoita. Siinä vaiheessa kun etäisyys lähettäjän ja vastaanottajan välillä kasvaa niin suureksi, että LAN- tai MAN-verkkojen kautta ei enää pystytä tyydyttävää yhteyttä muodostamaan, joudutaan hyödyntämään niin sanottuja suuralueverkkoja (WAN). Suuralueverkot rakentuvat yleensä telelaitosten toimittamista kiinteistä, tai osittain kytkentäisistä yhteyksistä, joiden datasiirtonopeudet liikkuvat tällä hetkellä 64 kbit/s ja 155 Mbit/s välillä. Nopeampia yhteyksiä on jo suunnitteilla/rakenteilla. Tässä kappaleessa on lyhyesti käyty läpi yleisimmät yhteyksien muodostustekniikat eli ISDN ja ATM. Näitä tekniikoita hyödyntäviä kaupallisia palveluita on markkinoilla runsaasti, mutta niitä ei katsottu tarpeelliseksi käsitellä tarkemmin tämän diplomityön yhteydessä.

### 3.3.1 ISDN

ISDN:llä tarkoitetaan digitaalista monipalveluverkkoa, jota ollaan kehittämässä digitaalisesta puhelinverkosta täydentämällä sitä ISDN-ominaisuuksilla kansainvälisten suositusten ja kansallisten standardien mukaan. ISDN-yhteys voi olla joko kytkentäinen tai kiinteä. Yleisimmät tällä hetkellä käytössä olevat ISDN yhteysratkaisut ovat

- piirikytkentäinen 64 kbit/s,
- piirikytkentäinen 3,1 kHz ja
- pakettivälitteinen 144 kbit/s.

Piirikytkentäinen 64 kbit/s verkkopalvelu tarjoaa käyttäjälle päästäpäähän läpinäkyvän siirtotien, joka ei aseta mitään vaatimuksia siirrettävän digitaalisen informaation sisällölle. Edellytyksenä on, että käyttäjä on kytkeytynyt digitaaliseen puhelinverkkoon. Piirikytkentäinen 3,1 kHz yhteys toteutetaan normaalin analogisen puhelinverkon kautta eli tästä johtuu rajallinen käytössä oleva kaistanleveys 3,1 kHz (taajuusalue 300-3400 Hz). Pakettivälitteinen 144 kbit/s on ISDN-liittymiä tarjoavien tahojen erikoispalvelu ja se toteutetaan yleensä kiinteänä yhteytenä. Tällä hetkellä ISDN-liittymät hyödyntävät jo olemassa olevia paketti-, puhelin- ja erillisverkkoja. Tavoitteena on vähitellen rakentaa itsenäinen ISDN-verkko, joka tulee tarjoamaan kaikki edellä mainittujen verkkojen palvelut.

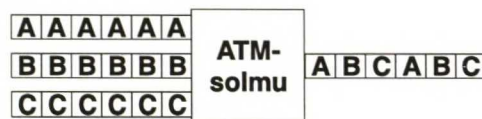


Käyttäjän ja ISDN-yhteyden välisessä rajapinnassa informaation ja merkinannon siirtoon käytetään erityyppisiä kanavia. Näitä kutsutaan B- ja D-kanaviksi. B-kanavia käytetään informaation siirtoon ja D-kanavia esimerkiksi erilaiseen merkinantoon. Kanavista voidaan muodostaa erilaisia yhdistelmiä ja käytössä olevat kanavat määräävät yhteyden siirtokapasiteetin. *Perusliitäntä* (N-ISDN, Narrowband Integrated Services Digital Network) muodostuu kahdesta 64 kbit/s B-kanavasta ja yhdestä 16 kbit/s D-kanavasta (2B+D) eli kokonaissiirtokapasiteetiksi muodostuu 144 kbit/s. *Järjestelmäliitäntä* (B-ISDN, Broadband Integrated Services Digital Network) muodostuu kolmestakymmenestä B-kanavasta ja yhdestä 64 kbit/s D-kanavasta (30B+D) eli kokonaissisäisyyden muodostuu 2 Mbit/s. B-ISDN:n standardointi on vielä kesken, mutta sen kytkentätekniikaksi on valittu jäljempänä kuvattu ATM.

ISDN-perusliittymään voidaan kytkeä samanaikaisesti kahdeksan "laitetta", joista kahta (2B) voidaan hyödyntää samanaikaisesti. Datasiirtonopeus kummallakin yhteydellä on mainittu 64 kbit/s. Sana "laite" on tässä tapauksessa hieman hämäävä, koska ISDN-liittymään voidaan kytkeä esimerkiksi kokonainen lähiverkko, jota käsitellään yhtenä "laitteena". Muita mahdollisia kytkettäviä laitteita voivat olla esimerkiksi yksittäiset tietokoneet tai puhelimet. ISDN-liittymästä muodostetaan yhteys palvelun tarjoajan keskukseseen, jonka kautta edelleen voidaan tarpeen mukaan hyödyntää piiri- tai pakettikytkentäisiä dataverkkoja, puhelinverkkoa, tietokantoja jne. Suomessa ISDN-liittymiä välittävät esimerkiksi Tele ja alueelliset teleyritykset. /18,42/

### 3.3.2 ATM

ATM eli Asynchronous Transfer Mode on ITU:n (International Telecommunication Union), ATM Forum ja ANSI:n (American National Standards Institute) standardoima nopea soluvälitykseen perustuva tiedonsiirtotekniikka. ATM siirron ensimmäisessä vaiheessa lähetettävä data pilkotaan 48 tavun mittaisiksi informaatiokentiksi, joihin jokaiseen liitetään 5 tavun mittainen otsikkokenttä. Kaikki liikenne, myös merkinantoinformaatio siirretään tämän tyyppisiin soluihin pakattuna. ATM-siirto tapahtuu päästä-päähän (point-to-point) muodostettuja virtuaaliyhteyksiä pitkin. Solujen reititys ATM-solmuissa tapahtuu solun otsikon sisältämän virtuaaliyhteyden tunnuksen mukaan. Tunnusta voidaan tarpeen mukaan vaihtaa eli se ei sisällä mitään todellista osoitetta, vaan ainoastaan yhteyden tunnisteen. Sana asynkroninen viittaa siihen, että kanavoitaessa ATM-yhteyksiä käytettävälle siirtotielle, samalla yhteydellä siirretyt solut voivat sijoittua täysin epäsäännöllisesti siirrettävään tietovirtaan. Asynkronisesta siirtotekniikasta on myös se hyöty, että kytkettävien yhteyksien ei tarvitse olla nopeudeltaan samantasoisia. /10,42/ Kuvassa 5 on esitetty periaate yksinkertaisena kaaviokuvana.



**Kuva 5.** ATM-solmun toimintaperiaate. Vasemmalta solmuun saapuvat hitaammat tiedonsiirtoyhteydet lomitetaan yhdelle nopeammalle yhteydelle. Yhden paketin pituus on aina määrämittainen eli 53 tavua.

ATM ei kuitenkaan ole vielä täysin valmis tekniikka kaiken tyyppisen datasiirron alustaksi. Ongelmia on tuottanut erityisesti pakettipohjainen datasiirto silloin, kun ATM-yhteyden kuormitus kasvaa riittävän suureksi (yli 90 %). Pakettipohjaisessa tiedonsiirrossa käytetään suhteellisen suuria pakettikokoja ja jos osa datasiirtoyhteydestä kulkee ATM-verkon kautta, joudutaan isokokoiset paketit pilkkomaan pieniksi soluiksi. Tästä on seurauksena,



että raskaasti kuormitetulla ATM-linjalla yhteen pakettiin kuuluvat solut saattavat fragmentoitua, jolloin koko paketti joudutaan lähettämään uudelleen. Seurauksena on verkon lisäkuormittuminen ja pakettivälitteisen liikenteen huomattava hidastuminen. Nykyisellään esimerkiksi IP-protokollaa käytettäessä, ATM-paketin koon tulisi olla noin kymmenkertainen jotta ongelmilta välttyttäisiin. Ratkaisuksi tähän ongelmaan ollaan kehittämässä ATM-solmuja, jotka jättävät koko paketin lähettämättä, jos fragmentoitumisen vaara on olemassa. Näin uudelleen lähetyksestä aiheutuva lisäkuorma voidaan välttää. /10/

## 3.4 Palvelimet

Palvelimet ovat yleensä tavallista tehokkaampia, mahdollisesti erikoisominaisuuksilla varustettuja työasemia, joten ne eivät sellaisenaan pysty tarjoamaan mitään palveluita verkkoon kytketyille laitteille ja sitä kautta käyttäjille. Käyttäjille näkyvät ja sovellusten hyödyntämät palvelut toteutetaan verkkokäyttöjärjestelmän ja muiden ohjelmistojen avulla. Verkkokäyttöjärjestelmän tarjoamiin palveluihin kuuluvat esimerkiksi juuri tämän työn kannalta olennaiset tulostuspalvelut. Verkkokäyttöjärjestelmällä on myös muita, yleensä tärkeämmiksi luokiteltuja tehtäviä kuin tulostimien ohjaus. Näitä ovat esimerkiksi:

- Tiedonsiirtopalveluiden tarjoaminen verkkoon kytketyille laitteille.
- Ohjelmistojen käyttäminen verkon kautta.
- Tallennuskapasiteetin tarjoaminen verkon kautta käyttäjille (verkkolevyt) jne.

Erillinen tulostinpalvelin voi olla tavallinen verkkoon kytketty työasema, joka on erilaisten tarkoitukseen laadittujen ohjelmistojen avulla valjastettu palvelemaan verkkoon kytkettyjä tulostimia ja niiden käyttäjiä. Koska ne huolehtivat ainoastaan verkkotulostuksen ohjauksesta ja hallinnasta, niiden kautta hoidettuna tulostaminen on tehokkaampaa ja tarjotut palvelut ovat tavallisesti huomattavasti monipuolisempia. Lisäksi tulostinpalvelin voi olla myös niin sanottu erillistuote eli vartavasten tarkoitukseen rakennettu tekninen laite. Seuraavassa on ensin lyhyesti käyty läpi tiedostopalvelimien tarjoamat peruspalvelut ja sen jälkeen varsinaisten tulostinpalvelimien ominaisuuksia.

### 3.4.1 Tiedostopalvelimet

Koska verkkokäyttöjärjestelmillä on runsaasti muita tehtäviä tulostimien ohjauksen lisäksi, ovat niiden tarjoamat tulostuspalvelut luonteeltaan hyvin perustavaa laatua. Seuraavassa on listattu palvelut, joihin lähes kaikki verkkokäyttöjärjestelmät pystyvät:

1. Tulostusjonot
  - a. Jokaista verkon kautta käytettävää tulostinta varten voidaan luoda oma tulostusjono.
  - b. Jokainen tulostettava tiedosto sijoitetaan tulostinta vastaavaan tulostusjonoon odottamaan prosessointivuoroa. Näin työasema, jolta tiedosto on lähetty vapautuu muuhun käyttöön.
  - c. Tulostettaville töille voidaan määrätä kiireellisyysaste, jolloin kiireelliset työt ohittavat muut jonossa olevat työt.
  - d. Töille voidaan asettaa tulostusajankohta. Tämä ominaisuus on hyödyllinen esimerkiksi erittäin suurten töiden kohdalla, jolloin ne voidaan määrätä tulostettavaksi vaikkapa työajan ulkopuolella.
2. Työnohjauspalvelut
  - a. Voidaan tarkastaa tulostimen tila eli onko työ kesken, onko tulostin valmis vastaanottamaan uusia töitä vai onko esimerkiksi paperi lopussa.

- b. Voidaan tarkastaa tulostusjonossa olevien töiden tila ja montako työtä jonossa on. Samalla nähdään töiden prosessointijärjestys.
- c. Voidaan keskeyttää tietyn työn käsittelyä ja jatkaa sitä jälleen halutulla hetkellä.
- d. Voidaan poistaa työ tulostusjonosta tai keskeyttää lopullisesti jo aloitettu työ.

### 3.4.2 Tulostinpalvelimet

Tulostinpalvelimiksi luokiteltavia ratkaisuja löytyy markkinoilta useita. Periaatteessa jako voidaan tehdä kahteen kategoriaan:

- ohjelmistopohjaisiin järjestelmiin ja
- laitteistopohjaisiin ("Hardware") järjestelmiin.

Ohjelmistopohjaisissa järjestelmissä palveluista huolehtii ohjelmisto, joka vaatii toimiakseen oman erillisen verkkoon liitetyn tietokoneen. Käytettävien tulostimien tulee lisäksi pystyä vastaanottamaan dataa verkon kautta eli käytännössä tämä tarkoittaa, että niiden tulee olla kytkettyjä joko suoraan palvelimeen (LPT/COM) tai verkkokortin avulla verkkoon. Tulostinpalvelinohjelmistojen tarjoamat palvelut käyttäjille ovat yleensä huomattavasti monipuolisempia kuin verkkokäyttöjärjestelmien vastaavat palvelut. Tämän tyyppiset ratkaisut sisältävät yleensä hyvin pitkälle kehitettyjä työnohjausominaisuuksia eli järjestelmän avulla pystytään valvomaan töiden käsittelyä ja tulostimia huomattavasti monipuolisemmin kuin esimerkiksi verkkokäyttöjärjestelmän tapauksessa. Seuraavassa on käyty läpi osa lisäominaisuuksista, joita löytyy ohjelmistopohjaisista tulostinpalvelimista:

- Tulostettavia töitä pystytään siirtämään tulostusjonosta toiseen. Järjestelmä voidaan konfiguroida tekemään tämä myös automaattisesti eli se hakee ensimmäisen vastaavilla ominaisuuksilla varustetun vapaana olevan tulostimen.
- Yhdestä tulostusjonosta voidaan tulostaa useammalle tulostimelle tai päinvastoin eli useista tulostusjonoista samalle tulostimelle.

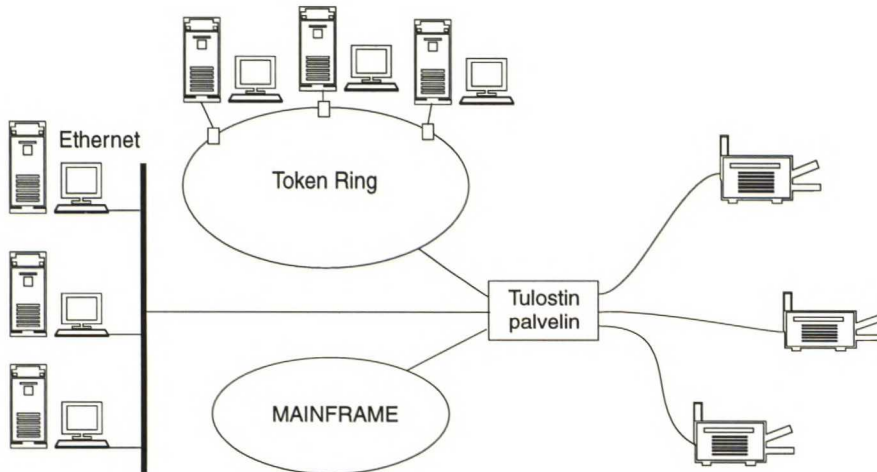
Laitteistopohjaiset ratkaisut muodostuvat yleensä pienestä verkkoon liitettävästä "laatikosta", johon voidaan liittää sarja- tai rinnakkaisporttien kautta 1-N tulostinta. Kaikki "äly" on laatikkoon sisäänrakennettua, eikä se yleensä ole millään tavoin päivitettävissä. Käytössä olevien porttien lukumäärä riippuu yleensä laitteen hinnasta eli halvimmat perusmallit sisältävät tietokoneiden liittimiä vastaavan konfiguraation, joka on yksi rinnakkaisportti (LPT) ja kaksi sarjaporttia (COM). Kalliimmissa malleissa portteja voi olla kymmeniä, mutta tällöin hinta on moninkertainen johtuen huomattavasti monimutkaisemmasta teknisestä toteutuksesta. Tämän tyyppiset laitteet tarjoavat muun muassa seuraavan tyyppisiä palveluita:

- Tulostusjonot ja niihin liittyvät peruspalvelut kaikille laatikkoon kytketyille tulostimille. Tulostusjonojen tapauksessa hyödynnetään usein verkkolevyjä tulostettavien töiden väliaikaisessa tallentamisessa.
- Kalliimmissa malleissa on mahdollista kytkeä palvelimeen useampia verkkoja, joiden ei välttämättä tarvitse käyttää samaa protokollaa. Tämän tyyppiset ratkaisut tarjoavat usein mahdollisuuden kytkeytyä myös "Mainframe"-järjestelmiin. (kuva 6).
- Kaikista palvelimeen kytketyistä verkoista voidaan tulostaa kaikille siihen kytketyille tulostimille samanaikaisesti.

Tämän tyyppisten ratkaisujen etu ohjelmistopohjaisiin tiedostopalvelin- ja tulostinpalvelinratkaisuihin verrattuna on hyvin pitkälle riippuvainen ympäristöstä jossa toimitaan. Laitteistopohjainen järjestelmä alentaa kuormitusta tiedostopalvelimella, mutta tähän perustuu myös erillisen tulostinpalvelimen käyttäminen. Lisäksi on huomattava, että tulos-



timiin ei tarvitse hankkia erillisiä verkkokortteja, vaan voidaan hyödyntää niiden perusliittimiä. Myöskään erillistä työasemaa ei tarvita, joten laitteistopohjaisen ratkaisun etuna voidaan pitää halvempaa hankintahintaa. Verkkokäyttöjärjestelmien ja laitteistopohjaisten järjestelmien tarjoamat palvelut eivät kuitenkaan pysty kilpailemaan ohjelmistopohjaisten ratkaisujen tarjoamien ominaisuuksien ja palveluiden kanssa.



**Kuva 6.** "Hardware"-pohjainen tulostinpalvelinratkaisu.

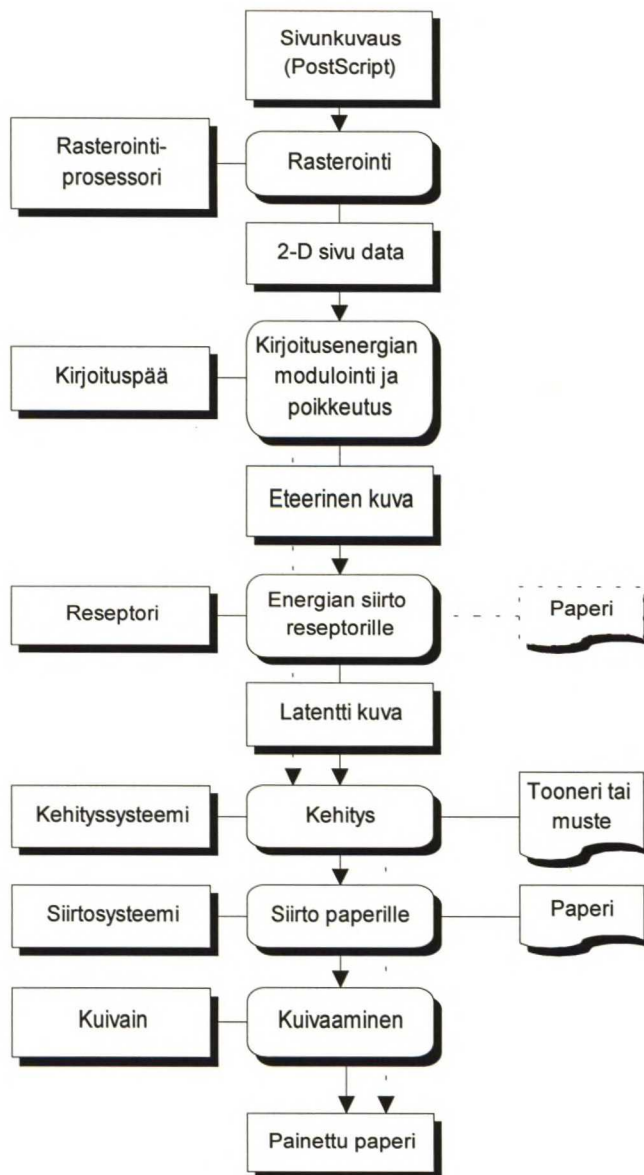
Verkkokäyttöjärjestelmä tarjoaa periaatteessa tuen rajattomalle määrälle tulostimia<sup>1</sup>, mutta mitä useampia tulostimia palvelee, sitä huonompi on suorituskky. Laitteistopohjaisten järjestelmien tapauksessa tulostimien määrä on aina rajallinen, koska käytössä on ainoastaan rajallinen määrä liittimiä jokaista palvelinta kohden. Ohjelmistopohjaiset tulostinpalvelinratkaisut tarjoavat tuen periaatteessa rajattomalle määrälle tulostimia samoin rajoituksin kuin verkkokäyttöjärjestelmät.

## 3.5 Tulostimet

### 3.5.1 NIP-tekniikat

Kuten jo aiemmin todettiin, elektronisissa painatusjärjestelmissä hyödynnetään NIP-tekniikoita. Näissä latentti kuva generoidaan uudelleen jokaista sivua varten eli jokainen sivu voi olla sisältöinformaatioltaan erilainen. Tämä NIP-tekniikoiden huomattava etu verrattuna konventionaalsiin painomenetelmiin on tietyllä tavalla myös sen heikkous. Latentin kuvan generoiminen uudelleen jokaista sivua varten hidastaa varsinaista tulostusprosessia huomattavasti: tästä syystä tämän tyyppistä järjestelmää ei voida pitää toiminnaltaan ideaalisena. Ideaalinen järjestelmä olisi sellainen, jossa latentti kuva luodaan reseptoripinnalle ainoastaan kerran, jonka jälkeen sitä voitaisiin käyttää halutun sivumäärän tulostamiseen. Tämän jälkeen reseptoripinta olisi mahdollista puhdistaa, generoida sille uusi latentti kuva ja jatkaa tulostamista. "Direct-to-plate" painatusjärjestelmissä ollaan jo kehittämässä tekniikkaa tähän suuntaan, mutta vielä tällä hetkellä reseptoripintana toimivat painolevyt ovat kertakäyttöisiä. NI-tulostuksen osavaiheet on esitetty kaavion muodossa kuvassa 7.

1. Rajoittavana tekijänä voidaan pitää verkkoon kytkettävien laitteiden maksimimäärää.



**Kuva 7.** Kuvan muodostuksen osavaiheet NI-painomenetelmissä. /39/

NI(Non-Impact)-menetelmissä latentti kuva voidaan muodostaa muun muassa seuraavilla mekanismeilla:

- Sähköisellä varauskuviolla, jota hyödynnetään elektrofotografiassa, elektrografiassa ja ionografiassa.
- Magnetoitumiskuviolla, jota käytetään magnetografiassa.
- Lämpökuviolla, jota hyödyntävät erilaiset termiset tulostusmenetelmät.
- Kemiallisella aktivoitumisella, jota käytetään niin sanotuissa mikrokapselimenetelmissä.

Oman ryhmänsä muodostavat inkjet-menetelmät, jotka eivät varsinaisesti hyödynnä mitään edellisistä mekanismeista, eikä niissä muodosteta myöskään kaksikulotteista latenttia kuvaa. Ne ovat niin kutsuttuja "non-contact"-tulostusmenetelmiä. Mustesuihku tulostusmenetelmiä on olemassa kahdenlaisia:



- CS (Continuous Stream) eli jatkuvan pisaroituksen menetelmiä ja
- DOD (Drop-On-Demand) eli epäjatkuvan pisaroituksen menetelmiä.

Elektrofotografiatekniikalla toteutetut tulostimet ovat tällä hetkellä ehdottomasti kilpailukykyisin vaihtoehto kun huomioidaan sekä tulostusnopeus että laatu. Suurin osa markkinoilla olevista elektronisen/hajautetun painatusjärjestelmän perustaksi sopivista tulostimista on toteutettu elektrofotografiatekniikalla. Markkinoilta löytyy kyllä laadullisesti parempia menetelmiä, jos kyseessä on esimerkiksi väritulostus (termografia), mutta niiden sivuntulostusnopeus on elektronista/hajautettua painatusjärjestelmää ajatellen aivan liian hidas. Vastaavasti löytyy elektrofotografiaa huomattavasti nopeampia menetelmiä (esimerkiksi magnetografia, joka voi olla noin 3-4 kertaa nopeampi), mutta niissä puolestaan laatu on yleensä huomattavasti heikompi. Mustesuihkutulostimet ovat vielä tällä hetkellä ensinnäkin liian hitaita ja toiseksi niiden laatua ei voida pitää riittävänä elektroniselle painatusjärjestelmälle. /9,35,36,39/

NIP-menetelmät voidaan jakaa epäsuoriin ja suoriin. Riippumatta menetelmästä ensimmäisessä vaiheessa rasterointiprosessori<sup>1</sup> tulkitsee sille syötetyn sivunkuvauksen ja muuntaa sen kaksidimensionaaliseksi sivulle siirrettäväksi pikseli-informaatioksi. Tässä vaiheessa suoritetaan myös esimerkiksi sävykuvien rasterointi. Tämän tietovirran ohjaamana tulostimen kirjoituspää moduloi kirjoitusenergiaa ja sijoittaa pikselit oikealle paikalleen. Epäsuorissa NIP-menetelmissä kuvainformaatio on eteerisessä eli aineettomassa muodossa siihen saakka, kunnes se on tallennettu erilliselle reseptoripinnalle. Tätä reseptoripinnalle tallennettua kuvaa kutsutaan latentiksi kuvaksi, joka kehitetään tavallisimmin toonerin avulla. Tämä kehitetty latentti kuva siirretään fyysisen kontaktin kautta paperin pinnalle. Lopuksi se vielä kuivataan tarttumisen varmistamiseksi. Suorissa NIP-menetelmissä ei käytetä erillistä reseptoria lainkaan, vaan reseptoripintana toimii paperi jolle latentti kuva muodostetaan suoraan. Tämän jälkeen kuva kehitetään kuten epäsuorien menetelmien tapauksessa. Suoraksi menetelmäksi luokitellaan myös esimerkiksi mustesuihkutulostuksen kaltaiset menetelmät, joissa kuva muodostetaan suoraan paperille. Näissä menetelmissä paperi toimii reseptorina, mutta kuvaa ei tarvitse varsinaisesti kehittää vaan paperille muodostetaan suoraan väri- tai tummuuskontrasti. /35,39/

### 3.5.2 Tulostimien luokittelu

Tulostimien luokittelussa yleisesti käytettäviä ominaisuuksia ovat

- toiminnallinen suorituskyky,
- määrällinen suorituskyky,
- laatu ja
- investointi- ja käyttökustannukset.

*Toiminnallisella suorituskyvyllä* tarkoitetaan tulostimen yleistä soveltuvuutta järjestelmän osaksi. Soveltuvuuteen voi vaikuttaa esimerkiksi, miten helposti tulostin on integroitavissa järjestelmään, mitä protokollia se tukee, miten sitä ohjataan, onko sen yhteyteen liitettävissä jälkikäsitteilyä, mitä ohjauskieltä/kieliä se ymmärtää ja niin edelleen. *Määrällisellä suorituskyvyllä* tarkoitetaan yksinkertaisesti tulostusnopeutta eli kuinka nopeasti tulostin pystyy tuottamaan valmiita sivuja. Vaikuttavia tekijöitä tässä tapauksessa ovat esimerkiksi rasterointiprosessorin teho ja tulostimen koneistonopeus. Tulostimet voidaan karkeasti tulostusnopeuden perusteella luokitella seuraaviin kategorioihin: *pienitehotulostimet* tuottavat alle 20 sivua minuutissa, *keskitehotulostimet* 20...60 sivua minuutissa ja *suurtehotulostimet* yli 60 sivua minuutissa. *Laadusta* puhuttaessa tarkoitetaan lopullisen tuotteen visuaalisia ominaisuuksia eli esimerkiksi täyspeittävien pintojen tasaisuutta ja sävykuvien

1. Käsitelty tarkemmin kappaleessa 3.6.



rasteroinnin onnistumista. *Investoinnistakustannuksiksi* voidaan lukea tulostimen hankintahinta. Lisäksi kysymykseen tulevat erilaiset järjestelmään liittämistä aiheutuneet kustannukset. *Käyttökustannuksia* ovat varsinaisesta tulostamisesta aiheutuvat kustannukset (paperin ja toonerin kulutus), sekä erilaiset huolto- ja ylläpitokustannukset. Ne ilmoitetaan yleensä hintana sivua kohden (p/sivu tai mk/sivu).

Vaikka tässä työssä tarkastelun kohteena on ainoastaan järjestelmän määrällinen suorituskyky, on tulostusjäljen laatu yksi tärkeimmistä kriteereistä tulostimia luokiteltaessa. Lisäksi laadulla on suora kytkentä järjestelmän suorituskykyyn, koska tärkeimpiä NIP-tulostuksen laatuun vaikuttavista tekijöistä on resoluutio. Resoluutio määritellään pikselin nimelliskoon käänteisarvona ja sen yksikkönä käytetään yleensä linjaa/tuuma (l/"), pistettä/tuuma (dpi) tai linjaa/senttimetri (l/cm). Mitä korkeampi resoluutio, sitä parempi laatu, mutta myös sitä enemmän pikseleitä tulostettavaa sivua kohden. Yleisimmin elektronisissa painatusjärjestelmissä käytetyt resoluutiot liikkuvat 300 ja 600 dpi:n välillä. Markkinoilta löytyy myös NIP-tulostimia, joiden resoluutio on 1200 dpi:tä ja ylikin, mutta niiden tulostusnopeudet ovat yleensä aivan liian alhaisia, jotta ne soveltuisivat käytettäväksi elektronisissa painatusjärjestelmissä.

Digitaalinen tulostus tapahtuu pikseleittäin, joko pikseli kerrallaan skannaamalla eli sarjamuotoisena tai rivi kerrallaan eli rinnakkaisena. Pikseleiden lukumäärä sivulla voidaan laskea kertomalla sivun koko resoluution neliöllä eli kaavan muodossa

$$\text{pikseleiden lukumäärä sivulla} = \text{sivun pinta-ala} \times \text{resoluutio}^2. \quad (1)$$

Kaavasta 1 voidaan edelleen johtaa yhtälö sivun tulostusnopeuden laskemiseksi, jos tunnetaan yhden pikselin tulostamiseen kuluva aika (Kaava 2). /35,39/

$$\begin{aligned} \text{sivun tulostusnopeus} &= \frac{\text{pikseleiden tulostusnopeus}}{\text{pikseleiden lukumäärä sivulla}} = \\ &= \frac{1}{\text{sivun pinta-ala} \times \text{resoluutio}^2} \times \frac{1}{\text{yhden pikselin tulostamiseen kuluva aika}} \end{aligned} \quad (2)$$

## 3.6 Rasterointiprosessori

Postscript-tulostinta ohjaavan rasterointiprosessorin eli RIPin (Raster Image Processor) tärkein osa on tulkki, joka nimensä mukaisesti tulkitsee sille syötetyn Postscript-kielisen ohjelman ja tiettyjen standardissa määriteltyjen sääntöjen mukaisesti suorittaa sen. Nämä säännöt määräävät esimerkiksi sen, missä järjestyksessä operaatiot suoritetaan ja miten Postscript-ohjelman osat sovitetaan yhteen halutun lopputuloksen saavuttamiseksi. Aiemmin rasterointiprosessoreiden toiminta oli hyvin suoraviivaista ja yksinkertaista. Komennot suoritettiin yksi kerrallaan syntaksin määräämässä järjestyksessä ja lopputuloksena saatiin valmis sivu. /7/ Huolimatta siitä, että prosessoreiden laskentateho on kasvanut huomattavasti, samalla myös sivunkuvauksen datamäärät ovat kasvaneet räjähdysmäisesti. Tämä johtuu monesta seikasta, mutta suurimpia syitä ovat väritulostuksen lisääntyminen ja tulostimien resoluution kasvaminen. Rasteroinnin tehokkuutta on pyritty lisäämään kehittämällä uusia prosessoriarkkitehtuureja.

Rasterointiprosessoriarkkitehtuurien perusmallina voidaan pitää systeemiä, joka rakentuu yhden suorittimen ympärille (CPU eli Central Processing Unit). Tämä rakenne on yleisesti käytössä lähes kaikissa tällä hetkellä markkinoilla olevissa perustulostimissa riippumatta siitä, mihin kategoriaan ne luokitellaan suorituskykynsä perusteella. Tämä johtuu hyvin pitkälle siitä, että tähän asti laskentatehoa on pystytty tehotarpeen kasvaessa lisäämään kehittämällä prosessoreita. Vasta viime aikoina on markkinoille ilmestynyt useamman pro-



sensorin ympärille rakentuvia arkkitehtuureja, jotka hyödyntävät niin sanottuja "pipeline"-tekniikoita tai rinnakkaislaskentaa. Niitä on hyödynnetty lähinnä korkearesoluutioisissa filmitulostimissa ja vasta muutama valmistaja on ryhtynyt soveltamaan kyseisiä tekniikoita alemman teholuokan tulostimiin. Itse asiassa "pipeline"-rasterointiarkkitehtuureja löytyy jo markkinoilta keskitehokuokan tulostimiin integroituina. Oletettavaa on, että lähitulevaisuudessa uusia RIP-arkkitehtuureja ryhdytään hyödyntämään yhä enemmän elektronisen painatuksen yleistyessä.

"Pipeline"-tekniikassa hyödynnetään joko useampaa prosessoria sarjassa tai sitten prosessoreiden moniajo-ominaisuuksia. Käytännössä tämä tarkoittaa, että rasterointiprosessorin suorittamat tehtävät jaetaan toisiaan ajallisesti seuraaviksi työvaiheiksi, joille jokaiselle osoitetaan oma suoritin tai lohkaistaan osa prosessorin tehosta. Kaikki toiminnot voidaan suorittaa päällekkäisinä eli samanaikaisesti voidaan vastaanottaa yhtä sivua, rasteroida toista ja tulostaa kolmatta. Yleensä jako tehdään seuraaviin työvaiheisiin:

- Uuden rasteroitavan sivun vastaanottaminen ja valmistaminen rasterointia varten.
- Varsinainen rasterointi eli koodin tulkitseminen ja muuntaminen pikseli-informaatioksi.
- Kirjoittimen koneiston ohjaaminen.

Rinnakkaislaskentaa hyödyntävät järjestelmät ovat vielä moninverroin tehokkaampia, koska niissä rasteroinnissa eniten resursseja kuluttava laskenta on jaettu useammalle prosessorille, jotka kaikki suorittavat saman työn rasterointia samanaikaisesti. Ongelmana on lähinnä ollut käytettyjen sivunkuvauskielten asettamat rajoitukset, eli sivunkuvauksen jakaminen erikseen käsiteltäviin palasiin ei aina ole mahdollista. Postscript Level 2 ja Adoben julkaisemat dokumenttien rakenne suositukset ovat vähitellen parantamassa tilannetta Postscriptin osalta. Adoben toinen sivunkuvauskieli PDF (Portable Document Format) soveltuu itse asiassa jo nyt rinnakkaisprosessointiin eli sen sivut ovat täysin toisistaan riippumattomia ja niitä voidaan käsitellä omina kokonaisuuksinaan.

Rasterointiprosessoreiden toteutustekniikoita on tänä päivänä käytössä useita. Jako voidaan tehdä laitteistopohjaisiin ja ohjelmistopohjaisiin rasterointiprosessoreihin. Vielä tällä hetkellä yleisin tekninen ratkaisumalli on integroida laitteistopohjainen rasterointiprosessori tulostimen yhteyteen. Sen etuna voidaan pitää sen täydellistä yhteensopivuutta tulostimen kanssa, mutta samalla RIPin päivittäminen on käytännössä mahdotonta. Nykyään löytyy monilta valmistajilta jo tulostimia, joihin on mahdollista vaihtaa kehittyneempi rasterointiprosessori laitteistopäivityksenä, eli käytännössä piirikorttia vaihtamalla. Uusinta teknologiaa edustavat niin sanotut SLB(Soft Loadable System)-järjestelmät, joissa tulostimen RIP on päivitettävissä esimerkiksi ohjelmistoa tai rasterointialgoritmia vaihtamalla. Ero varsinaisiin ohjelmistopohjaisiin RIPpeihin piilee siinä, että algoritmi ladataan edelleen tulostimen yhteyteen integroidulle piirikortille, eikä esimerkiksi erilliselle työasemalle tai tulostimen massamuistiin.

Rasterointiprosessori voidaan myös liittää osaksi työasemaa, jolloin tietokoneen lisälaitteporttiin liitetään rasterointikortti, joka sisältää oman prosessorinsa. Kokonaan oman ryhmänsä muodostavat rasterointiohjelmat, jotka ovat täysin ohjelmistopohjaisia eli ne vaativat alustakseen normaalin työaseman ja hyödyntävät suoraan sen prosessoria. Tämän tyyppisiä rasterointiohjelmia toimitetaan myös valmiina työasemapaketteina, jolloin työasemat yleensä viritetään tukemaan ohjelmiston ominaisuuksia parhaalla mahdollisella tavalla, ja niihin saatetaan lisätä esimerkiksi erilaisia apuprosessoreita. On myös rakennettu järjestelmiä, jotka sisältävät useampia rinnakkaisia, toisistaan riippumattomia rasterointiprosessoreita. Niiden yhteyteen on useimmiten liitetty myös massamuisti, valmiiksi rasteroitujen töiden puskuriksi. Tulostin hakee seuraavana jonossa olevan valmiiksi rasteroidun työn muistista kun se on saanut edellisen tulostettua. Tämän tyyppiset rasterointiprosessorit pystyvät yleensä ohjaamaan useampia tulostimia samanaikaisesti.

/1,4,5,6,14,15,16,17,46,47/

# 4

## MALLINNUSTEKNIIKAT

Tässä luvussa käydään ensin lyhyesti läpi yleisiä piirteitä oppivista ja älykkäistä järjestelmistä ja pyritään valottamaan niiden toimintaperiaatteita sekä sovelluskohteita. Menetelmien välillä on tässä yhteydessä pyritty suorittamaan vertailua, jotta pystyttäisiin paremmin perustelemaan, miksi hajautetun painatusjärjestelmän mallinnustekniikaksi lopulta valittiin juuri neurolaskenta. Tämän jälkeen paneudutaan tarkemmin neurolaskennan periaatteisiin, yleisimpiin menetelmiin ja matemaattiseen taustaan. Varsinaista mallia ja sen kehittelyyn liittyvää problematiikkaa ei tässä luvussa käsitellä, tavoitteena on ainoastaan luoda yleiskuva neurolaskennasta, ei esittää ratkaisumallia tiettyyn spesifiseen ongelmaan. Tästä syystä tekstissä on pyritty pitäytymään hyvin yleisessä teoriassa. Varsinaisen mallin kehittämiseen palataan diplomityön kokeellisessa osassa luvussa 6.

### 4.1 Oppivat ja älykkäät järjestelmät

#### 4.1.1 Määritelmä

Oppivilla ja älykkäillä järjestelmillä tarkoitetaan tietojenkäsittelyjärjestelmiä, joissa pyritään jäljittelemään keinotekoisesti ihmisen tapaa käsitellä asioita. Kyseiset järjestelmät voidaan jakaa karkeasti kahteen kategoriaan: laskennallisiin järjestelmiin ja tekoälysovelluksiin. Laskennallisesti älykkäät järjestelmät käsittelevät informaatiota ainoastaan numeerisessa muodossa, kun taas tekoälysovelluksissa asioille ja olioille pyritään antamaan symbolisia nimiä. Reaalimaailmaa mallinnettaessa juuri tämä piirre tekee laskennallisista järjestelmistä ylivoimaisia verrattuna tekoälyyn, koska fysiikan lakien ja logiikan sääntöjen sijasta voidaan käyttää mittauksista ja tilastoista saatua dataa. Voidaan sanoa, että laskennallisissa järjestelmissä tunnustetaan reaalimaailman monimutkaisuus ja sen ideaalisen mallintamisen mahdottomuus. /27/

#### 4.1.2 Yleisimmät menetelmät

Tällä hetkellä kolme tunnetuinta menetelmää oppivista ja älykkäistä järjestelmistä puhuttaessa ovat sumeat järjestelmät, neurolaskenta ja geneettiset algoritmit. Näistä sovellusten kannalta selkeimpiä lienevät sumeat järjestelmät, neurolaskenta vaatii jo hieman enemmän teoreettista tietämystä ja geneettiset algoritmit ovat vasta tulossa sovellusasteelle. /27/ Niihin jokaiseen tutustutaan seuraavassa hieman tarkemmin. Neurolaskentaa käsitellään tässä yhteydessä vain pintapuolisesti, koska siihen palataan huomattavasti tarkemmin kappaleessa 4.2.



## Sumeat järjestelmät

Sumeiden järjestelmien toiminta perustuu epätasämallisyyden tai epävarmuuden tarkasteleluun tähän tarkoitukseen erityisesti formuloidun logiikan ja joukko-opin avulla. Menetelmiä epätasämallisyyden käsittelemiseksi on ryhdytty kehittämään koska on havaittu, että suurin osa ympäröivän maailman ilmiöistä on enemmän tai vähemmän epätasämallisia. Tieteellisissä piireissä epätasämallisyyttä on aina pidetty välttämättömänä pahana ja niinpä se on aina pyritty kaikin mahdollisin keinoin eliminoimaan. Esimerkiksi tieteellisten termien on aina oletettu olevan yksiselitteisesti määriteltyjä, vaikka käytännössä tätä vaatimusta ei aina ole voitu noudattaa. Tältä pohjalta kehitetyissä malleissa epätasämalliset ilmaisut yleensä pyrittiin täsmällistämään, jolloin varsinkin tietokoneympäristössä saatiin liian pelkistettyjä ja yksinkertaistettuja teorioita ja malleja. Epävarmuutta puolestaan on perinteisesti käsitelty todennäköisyyden käsitteen avulla. /23/

Käytännössä sumean logiikan hyödyntäminen tarkoittaa siirtymistä kaksiarvologiikasta moniarvologiikkaan. Moniarvologiikalla tarkoitetaan, että totuusarvoja on useampia kuin kaksi eli muitakin kuin "tosi" tai "epätosi". Sumeiden systeemien teorian perustana on niin sanottu sumea joukko-oppi. Konventionaalisessa joukko-opissa objekti joko ei kuulu tai kuuluu joukkoon, eli se voi saada jäsenyysarvon 0 tai 1. Sumeassa joukko-opissa objekti voi kuulua joukkoon vain osittain. Tämän tyyppinen esitys toteutetaan määrittelemälle jokaiselle objektille jäsenyysaste, joka voi olla lukuarvo esimerkiksi väliltä  $[0,1]$  eli mitä pienempää osittainen kuuluminen joukkoon on, sitä lähempänä arvo on nollaa ja päinvastoin. Objekti voi näin ollen kuulua myös useampaan kuin yhteen joukkoon. Kun objekteille on määritelty niiden jäsenyysarvot, voidaan niiden perusteella suorittaa erilaisia päättelyitä, jotka poikkeavat perinteisten menetelmien päättelystä huomattavasti. Sumean logiikan avulla voidaan esimerkiksi esittää seuraava lause matemaattisesti:

jos huone on melko kylmä, niin lämmitä melkein täydellä teholla.

Perusedellytys käsittelylle on, että sanoille *melko* ja *melkein* on määritelty niitä kuvaavat vaihteluvälit. /23/

## Neurolaskenta

Neurolaskenta on vanhin ja laajimmin tunnettu laskennallisen älykkyyden menetelmistä. Neurolaskenta soveltuu parhaiten erilaisten suuresta informaatiomäärästä vaikeasti havaittavien lainalaisuuksien ja riippuvuuksien selvittämiseen ja ennustamiseen. Neurolaskennan periaate on matematiikaltaan hyvin yksinkertainen, joskin runsaasti laskentatehoa vaativaa. Laskenta suoritetaan niin sanotuilla neuroverkoilla, jotka muodostuvat neuroneista ja niitä yhdistävistä synapseista. Yhdellä neuronilla voi olla useita syötteitä, mutta ainoastaan yksi vaste. Neuronin syötteiden vaikutusta vasteeseen joko vahvistetaan tai heikennetään painokertoimien avulla. Vasteen lopullinen arvo summataan painotetuista syötteistä. Yleensä neuroni sisältää myös niin sanotun aktivaatiofunktion, jonka avulla voidaan vaikuttaa vasteen esitysmuotoon. Neuroverkkojen toimintaa on lähes mahdoton ymmärtää pelkkien painokertoimien kautta ja tästä syystä painot määrätään yleensä esimerkeistä opettamalla. Erilaisia verkkomalleja ja opetusalgoritmeja on olemassa runsaasti. /23,27,31/

## Geneettiset algoritmit

Geneettiset algoritmit ovat uusin tulokas oppivien ja älykkäiden järjestelmien joukossa, ja kuten aiemmin jo mainittiin, ne ovat vasta tulossa sovellusasteelle. Niiden toiminta perustuu yrityksen ja erehdyksen hyödyntämiseen ongelmanratkaisussa eli olemassa olevasta ratkaisusta kehitellään erilaisia muunnoksia joita testataan, ja näiden testien perusteella hylätään huonosti toimivat vaihtoehdot. Tämän jälkeen hyviksi todetuista muunnoksista kehitellään uusia muunnoksia, jotka testataan ja niin edelleen. Menetelmä on saanut nimensä siitä, että muunnoksia generoivia algoritmeja luotaessa on mallina käytetty geneetikasta tuttuja mutaatioita ja geenimuutoksia aiheuttavia mekanismeja. Menetelmässä on

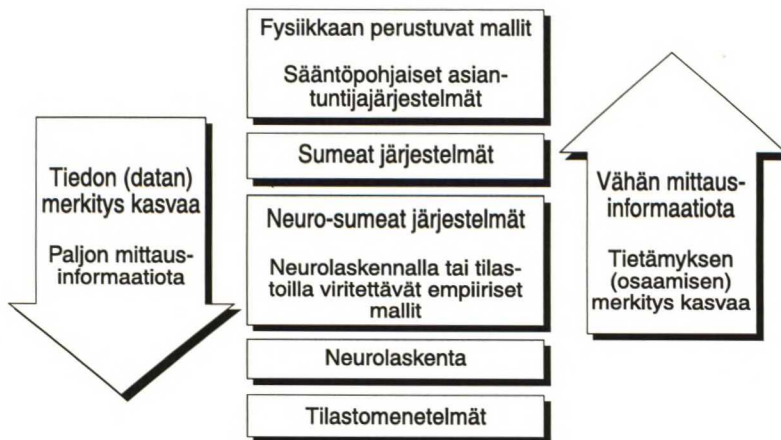


siis itse asiassa kyse optimointitekniikasta, joka useiden neurolaskennan menetelmien tavoin luetaan kuuluvaksi satunnaishakumenetelmiin. /27/

#### 4.1.3 Menetelmien vertailu

Sumea logiikka perustuu JOS-NIIN tyyppisten sääntöjen hyödyntämiseen samalla tavoin kuin esimerkiksi asiantuntijajärjestelmissä. Asiantuntijajärjestelmien ja sumeiden järjestelmien ero piilee siinä, että sumean logiikan avulla säännöt voidaan yleistää kattamaan suurempi joukko tapauksia ja lisäksi sen avulla voidaan hallita sääntökannan ristiriitaisuuksia. Koska konventionaaliset joukot ovat sumeiden joukkojen erikoistapauksia, voidaan periaatteessa mikä tahansa systeemi sumeuttaa. Tästä syystä sumeaa logiikkaa on sovellettu useilla tutkimuksen aloilla, joista tyypillisimpiä ovat olleet säätöteoria, hahmon tunnistus, diagnostiikka, päätösteoria, optimointiteoria, robotiikka, "älykkäät järjestelmät", sosiologia ja käyttäytymistieteet. Tähän mennessä menestyksekkäin sumean logiikan sovel-lusalue on ollut säätötekniikka ja suurin osa kaupallisista sovelluksista on erilaisia säätö-järjestelmiä sivuavia. /23/

Neuroverkot eivät suoraan käsittele JOS-NIIN-sääntöjä, kuten edellä mainitut asiantunti-jajärjestelmät ja sumeat järjestelmät, vaan ne oppivat säännöt itse. Säännöt opetetaan niille kuvattavan järjestelmän syöte-vaste parien avulla. Yksi haittapuoli tekniikalla kui-tenkin on: koska säännöt esitetään verkon muodostavien solmujen painokertoimina, on täl-laisen mallin hahmottaminen ihmiselle huomattavasti hankalampaa kuin JOS-NIIN-sääntöjen. Tämä voi aiheuttaa ongelmia järjestelmää viritettäessä. Neuroverkkojen yksi suurimmista eduista puolestaan on, että niille ei tarvitse opettaa kaikkia mahdollisia järjes-telmän syöte-vaste pareja. Ne siis pystyvät opettamisen jälkeen ennustamaan myös ennen-kokemattomia tilanteita. Tästä on seurauksena, että kun halutaan mallintaa erittäin hankalia järjestelmiä, joissa syy-seuraus suhteet ovat erittäin monimutkaisia on mittausten avulla saatua opetusmateriaalia oltava runsaasti. /23,27/



**Kuva 8.** Menetelmien soveltuvuus erityyppisten järjestelmien mallintamiseen siinä tapa-uksessa, että soveltuvuutta mitataan käytössä olevan, järjestelmää kuvaavan informaation määrällä.

Viime aikoina on tutkittu runsaasti menetelmiä, joissa hyödynnetään sekä neurolaskentaa, että sumeaa logiikkaa. Näissä järjestelmissä voidaan esimerkiksi sumeaa järjestelmää virit-tää neuroverkkojen avulla. Kyseisiä järjestelmiä on pidetty erittäin lupaavina siitä syystä, että ne yhdistävät sumeiden järjestelmien esityksellisen selkeyden ja neuroverkkojen oppi-miskyvyn. Markkinoilta löytyy tällä hetkellä vain muutamia sovelluksia ja ne ovat toteu-



tukseltaan hyvin alkeellisia; ne keskittyvät lähinnä vain tietyn kapean osa-alueen ongelmiin. /23/ Kuvassa 8 on kuvaajan muodossa pyritty esittämään mallinnusmenetelmien suhdetta toisiinsa.

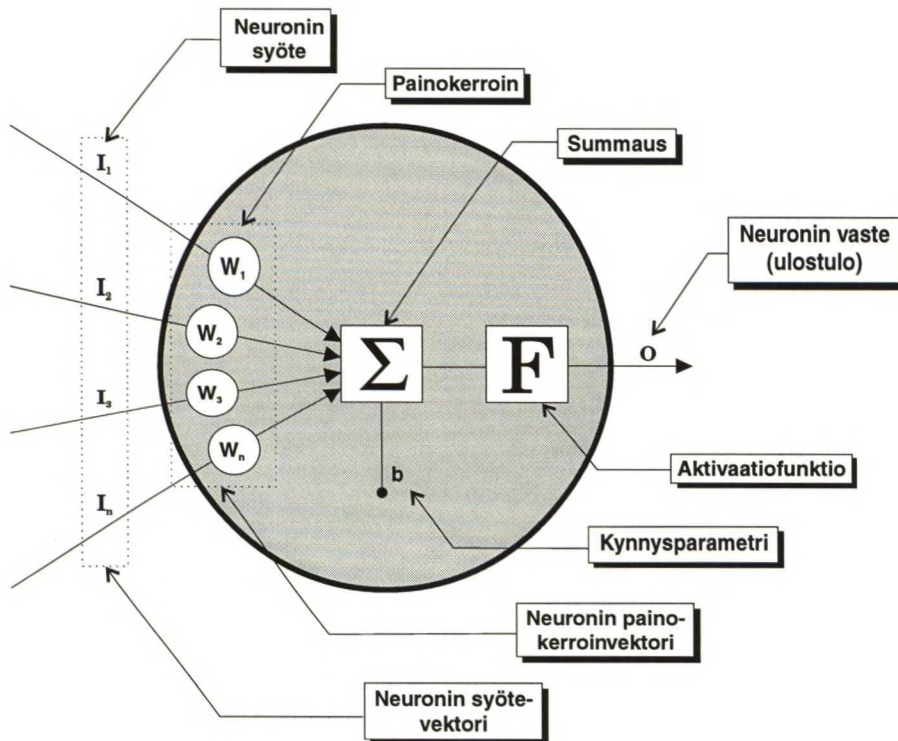
## 4.2 Neurolaskenta

### 4.2.1 Periaatteet ja matemaattinen tausta

Neuronimalli ja neuroverkon arkkitehtuuri kuvaavat neuroverkkojen tapaa muuntaa syötteensä vasteiksi. Tätä muunnosta voidaan tarkastella yksinkertaisina laskutoimituksina. Neuronimalli ja verkon arkkitehtuuri asettavat tarkat rajat sille, mitä neuroverkon avulla voidaan kuvata.

#### Neuronimalli

Yksinkertaisimmillaan neuroni muodostuu syötteestä, sitä vastaavasta painokertoimesta, aktivaatiofunktioista ja vasteesta. Vaste lasketaan kertomalla syöte painokertoimella ja muokkaamalla tätä tulosta aktivaatiofunktioilla. Hieman monimutkaisempi rakenne sisältää niin sanotun kynnysparametrin (*bias*), jonka tehtävänä on siirtää aktivaatiofunktion toimipistettä. Siirtäminen tapahtuu summaamalla kynnysparametri syötteen ja painokertoimen tuloon ennen aktivaatiofunktion syöttämistä. Painokerrointa ja kynnysparametriä käsitellään muuttujina eli niiden arvoja muuttamalla pyritään vaikuttamaan neuronin toimintaan halutun lopputuloksen saavuttamiseksi. Painokertoimen ja kynnysparametrin arvoja voidaan muuttaa käyttäjän toimesta tai verkko voidaan suunnitella siten, että se pyrkii itse optimoimaan nämä arvot parhaan mahdollisen tuloksen saavuttamiseksi.

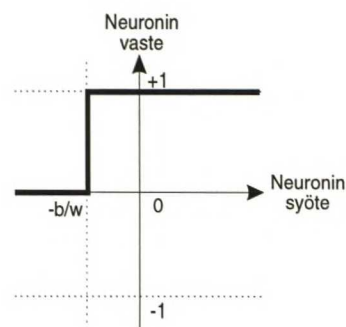
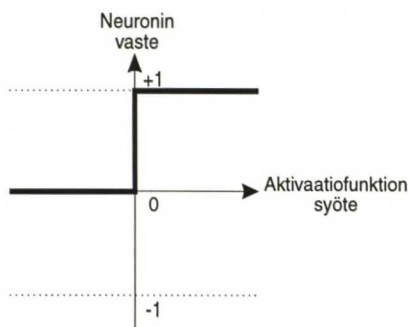


Kuva 9. Neuronin rakenne.

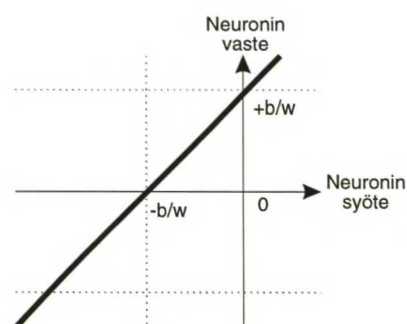
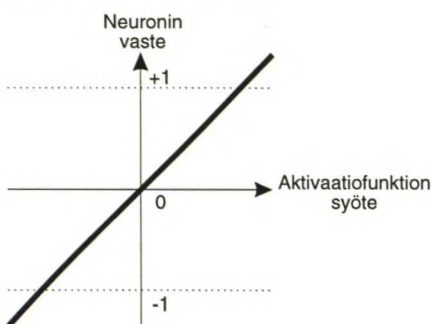
Todellisissa sovelluksissa jokaisella neuronilla on yleensä useampi kuin yksi syöte. Kuvassa 9 on kuvattu tämäntyyppisen neuronin rakenne ja toimintaperiaate. Kuten kuvasta havaitaan neuronilla on tässä tapauksessa useampia syötteitä ( $I_1 \dots I_n$ ), joista jokaisella on oma painokerroin ( $W_1 \dots W_n$ ). Näistä lukuarvoista muodostetaan vektoreita, joihin voidaan soveltaa normaaleja vektorilaskusääntöjä. Usean syötteen neuronin toimii periaatteessa täsmälleen samalla tavoin kuin edellä kuvattu yhden syötteen neuronin. Ainoa ero on, että syötteiden ja painokertoimien tulot lasketaan yhteen ennen kynnysparametrin summamista ja aktivaatiofunktioille syöttämistä. Matemaattisesti ajateltuna kysymyksessä on siis vektoreiden  $I$  ja  $W$  pistetulo. Ainoastaan yhden vektorin syöttäminen kerrallaan neuroverkolle on suhteellisen hidasta ja tästä syystä syöte- ja vastevektorit yleensä kootaan matriiseiksi, joita voidaan käsitellä kokonaisuuksina ja joihin pätevät kaikki normaalit matriisilaskennan periaatteet. Osa neurolaskennan menetelmistä jopa edellyttää, että opetusaineiston vektorit esitetään niille matriisin muodossa, jotta oppimisalgoritmit pystyvät huomioimaan niiden väliset riippuvuudet.

Aktivaatiofunktion tehtävä on määrittää esitystapa neuronin ulostulolle. Aktivaatiofunktioita on olemassa useita erilaisia ja lisäksi neurolaskentasovellukset yleensä tarjoavat mahdollisuuden määrittellä käyttäjäpohjaisia funktioita. Seuraavassa on käyty läpi kolmen yleisimmin käytetyn aktivaatiofunktion periaatteet. Jokaisen kohdalla on esitetty funktion toimintaperiaate myös graafisesti. Kuten kuvaajista havaitaan, kaikissa kolmessa tapauksessa voidaan kynnysparametrin avulla muuttaa pistettä, johon aktivaatiofunktion syötteen arvoa verrataan. /12,13,21,26,28,31/

1. **Askelfunktion** ulostulo voi saada ainoastaan kaksi arvoa, jotka yleensä ovat joko 0 tai 1. Funktio käsittelee syötteen ja painokertoimen tulon hyvin yksinkertaisesti. Jos askelfunktiolle syötetty arvo on alle nollan, saadaan neuronin ulostuloksi arvo 0. Jos se puolestaan on suurempi tai yhtäsuuri kuin nolla, saadaan neuronin syötteelle arvoksi 1. Askelfunktiolle voidaan määrittellä myös muita ulostuloarvoja eli esimerkiksi -1 ja 1. Tämän tyyppiset aktivaatiofunktiot ovat erityisen käyttökelpoisia erilaisissa luokittelutehtävissä.

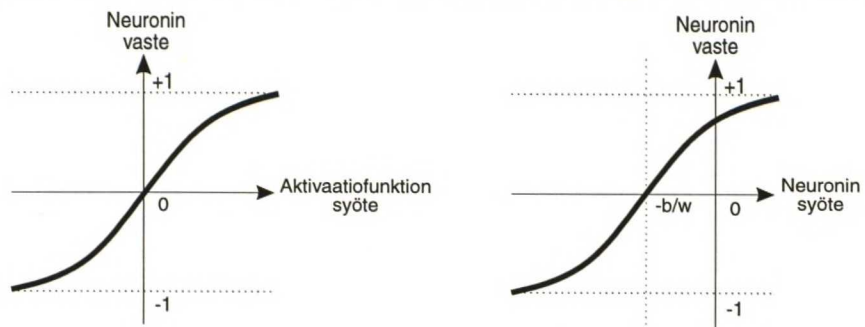


2. **Lineaarinen aktivaatiofunktio** ei vaikuta sille syötettyyn arvoon millään tavoin, vaan välittää sen sellaisenaan neuronin vasteeksi. Tämän tyyppistä aktivaatiofunktiota hyödynnetään järjestelmissä, joissa on tarpeen esittää vasteiden arvot  $-\infty$  ja  $+\infty$  välillä.



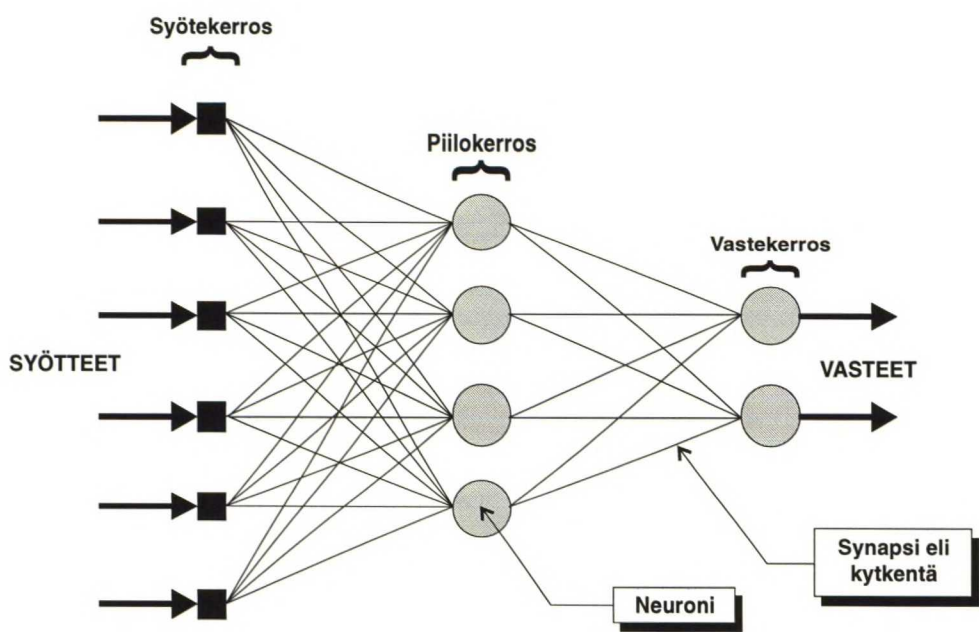


3. **Sigmoidifunktioita** käytetään yleisesti kahden tyyppisiä eli logaritmisia ja tangentiaalisia. Logaritmiset sigmoidifunktiot skaalaavat neuronin vasteet 0 ja 1 välille, kun taas tangentiaalisten sigmoidifunktioiden ulostulo voi sijoittua välille -1 ja 1. Alla on graafisesti esitetty tangentiaalinen sigmoid-aktivaatiofunktio.



Arkkitehtuuri

Käytännössä neuroverkot muodostuvat vähintään kahdesta, mutta yleensä useammasta yllä kuvatun kaltaisesta neuronista. Neuroneita voidaan sijoittaa verkkoon rinnakkain, jolloin muodostuu ikään kuin neuronikerroksia. Näitä kerroksia voidaan edelleen sijoittaa useampia peräkkäin, jolloin voidaan muodostaa erittäin monimutkaisia verkkoarkkitehtuurreja. Neuronien välille määritellään kytkentöjä eli niin sanottuja synapseja, joiden avulla pystytään yksiselitteisesti määrittämään kunkin neuronin syötteet ja laskettua vastetta edelleen syötteenään käyttävät neuronit. Kuvassa 10 on graafisesti esitetty yksinkertaisen neuroverkon rakenne. Se muodostuu kuuden syötteen syötekerroksesta ja kahdesta neuronikerroksesta. Verkon viimeistä neuronikerrosta kutsutaan yleensä vastekerrokseksi. Jos verkkorakenne sisältää vastekerroksen lisäksi muita neuronikerroksia kuten tässä tapauksessa, kutsutaan näitä puolestaan piilokerroksiksi. Vastekerroksessa on aina niin monta neuronkia kuin järjestelmässä on vasteita. Piilokerroksien neuronien lukumäärä sen sijaan ei ole millään tavoin riippuvainen syötteiden tai vasteiden määrästä ja se joudutaan aina määrittämään tapauskohtaisesti. /12,13,21,26,28,31/



**Kuva 10.** Yksinkertainen neuroverkko, jolla on kuusi syötettä, neljä piilokerroksen neuronia ja kaksi vasteneuronia. /27/

Neuroverkon arkkitehtuuri kertoo siis seuraavat asiat:

- Kuinka monta neuronikerrosta verkossa on.
- Kuinka monta neuronia kukin kerros sisältää.
- Mikä on kussakin kerroksessa hyödynnetty aktivaatiofunktio.
- Miten eri neuronikerrokset on kytketty toisiinsa.
- Montako syötettä ja vastetta neuroverkolla on.

### Oppimisalgoritmit

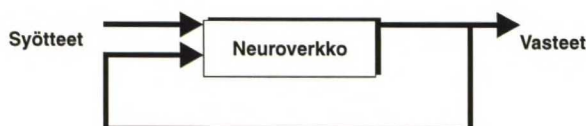
Oppimisalgoritmeilla tarkoitetaan joukkoa peräkkäisiä operaatioita verkon painokertoimien määrittämiseksi. Näitä operaatioita toistetaan koko verkon opetuksen ajan, ja ne määräävät miten oppiminen verkossa tapahtuu. Neuroverkkoa opetettaessa opetusaineisto joudutaan käymään läpi useita kertoja. Yhtä opetusjaksoa, jonka aikana koko aineisto on käyty kertaalleen läpi kutsutaan opetuskierrökseksi (Epoch). Usein oppimisalgoritmit ovat niin tiukasti sidoksissa tiettyyn seuraavassa kappaleessa kuvatuista neurolaskennan menetelmistä, että koko menetelmä on nimetty oppimisalgoritmin mukaan. Oppimisalgoritmit sisältävät yleensä erilaisia parametrejä, joiden avulla voidaan ohjata opetusta. Näistä tärkein on niin sanottu oppimisnopeus-parametri. Se esiintyy useimmissa neuroverkkoalgoritmeissa ja sen avulla säädetään painokertoimiin yhdellä kertaa tehtävän muutoksen suuruutta.

## 4.2.2 Neurolaskennan menetelmien luokittelu

Neurolaskennan menetelmät voidaan jakaa kolmeen ryhmään, jotka ovat *kiinteäpainoiset verkot*, *ohjatus oppimisen verkot* ja *itseoppivat verkot*.

### Kiinteäpainoiset verkot

Kiinteäpainoisissa verkoissa ei tapahdu oppimista vaan neuronien painot on määrätty ennaltakäsin. Kuvassa 11 on esitetty tämän tyyppisten verkkojen toimintaperiaate. Syötevektori ei kaikissa verkoissa ole välttämätön eli verkon alkutila voidaan määrätä myös neuronien vasteiden avulla. Kiinteäpainoisten verkkojen rakenteeseen kuuluu olennaisesti takaisinkytkentä. Verkon toimintaperiaatteena on, että neuronien ulostulot antavat lopputilassaan vain järkeviä vastauksia. Verkko toimii kuten tilakone, joka muutettuaan tilaansa useita kertoja jää lopulta painojen ja alkutilan määräämään stabiiliin lopputilaan, josta tehtävän vastaus voidaan lukea. Vastaukset voivat olla esimerkiksi muistiin tallennettuja hahmoja (vektoreita), jolloin puhutaan autoassosiatiivisista muisteista. Jos autoassosiatiivisen muistin alkutilaksi annetaan vääristynyt tai puutteellinen hahmo, niin verkko pyrkii löytämään muistista sitä parhaiten vastaavan esimerkin. Tämän tyyppisiä verkkoja voidaan käyttää myös optimointitehtävissä, jolloin verkko asettuu alkutilastaan johonkin sallittuun lopputilaan, jossa verkon yksiköiden arvoista voidaan lukea optimointitehtävän ratkaisu. /21,27/

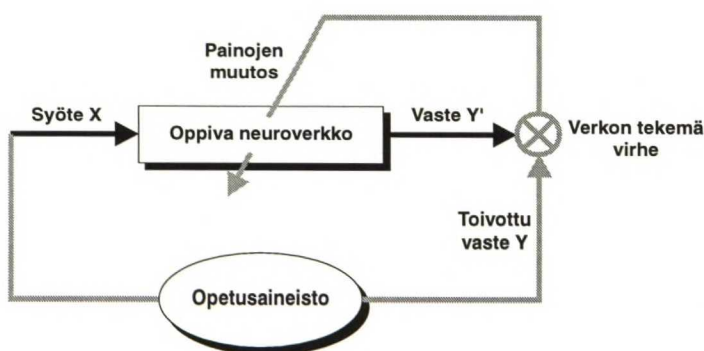


**Kuva 11.** Kiinteäpainoisten verkkojen toteutusperiaate. /27/



### Ohjatun oppimisen verkot

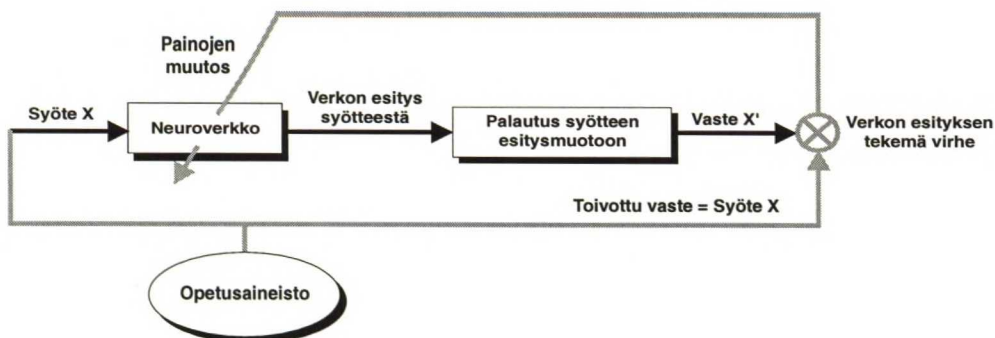
Ohjattu oppiminen on kaikkein suosituin neurolaskennan menetelmistä. Siinä neuroverkko opetetaan joukolla syöte-vaste-pareja, jolloin oppimisen tuloksena syntyy eräänlainen yleistävä laskentakaava, jonka avulla voidaan ennustaa miten tietyn tyyppiseen syötevektoriin tulee vastata. Yleistävällä laskentakaavalla tarkoitetaan, että neuroverkko antaa järkevän vastauksen myös sellaisen syötevektorin perusteella, joka ei ole sisällynyt opetusaineistoon, mutta jonka avulla vastaus on johdettavissa. Opettaminen tapahtuu valitsemalla aineistosta yleensä satunnaisesti syöte-vaste-pari (esimerkiksi syötevektori  $X$  ja sen vastevektori  $Y$ ). Verkon avulla lasketaan syötteestä  $X$  vaste  $Y'$ . Tämän jälkeen todellista vastetta  $Y$  ja laskettua vastetta  $Y'$  verrataan toisiinsa ja niiden välistä eroa käytetään verkon painojen korjaamiseen. Laskennallisesti opetus on yleensä niin hidas toimenpide, että tämän tyyppiset neuroverkot eivät sovellu reaaliaikaiseen mallintamiseen. Kuvassa 12 on esitetty opetusprosessi kaavion muodossa. /27,44/



**Kuva 12.** Ohjatun oppimisen periaate. /27/

### Itseoppivat verkot

Itseoppivien verkkojen tapauksessa tunnetaan syöte, mutta ei vastetta. Oppimisen tehtävänä on löytää annetuista syötteistä lainalaisuuksia, joita voidaan hyödyntää aineiston tulokinnassa tai esimerkiksi muiden neurolaskentamenetelmien syötteinä. Neurolaskentaa hyödynnetään siis itse asiassa syöteinformaation käyttäytymistä kuvaavan esityksen muodostamisessa. Oppimisalgoritmit on usein toteutettu siten, että neuronit ikään kuin kilpailevat siitä, mikä niistä edustaa kutakin syötettä parhaiten. Tästä syystä menetelmää kutsutaan usein myös kilpailuoppimiseksi. Kuvassa 13 on esitetty algoritmien toimintaperiaate kaavion muodossa. /27/



**Kuva 13.** Periaate, jolla itseoppivat verkot voidaan toteuttaa. /27/

### 4.2.3 Tunnetuimmat neurolaskennan menetelmät

Tässä kappaleessa on lyhyesti käyty läpi yleisimmät neurolaskennan menetelmistä. Niiden perinpohjainen esittely on mahdotonta, mutta teoriaa on pyritty käymään läpi lähinnä sitä silmällä pitäen, että pystyttäisiin perustelemaan kokeellisessa osassa mallien muodostusvaiheessa tehdyt arkkitehtuuri- ja algoritmivalinnat. Huomattavaa on, että osa alla kuvatuista menetelmistä voidaan edellisessä kappaleessa esitellyn luokittelun perusteella sijoittaa samaan kategoriaan: ainoa niitä erottava tekijä on käytetty opetusalgoritmi.

#### Perceptron

Perceptron on yksi ensimmäisistä neurolaskennan menetelmistä, jonka kehitti Yhdysvaltalainen F. Rosenblatt jo vuonna 1961. Verkko rakentuu yhdestä kerroksesta neuroneita, joiden kynnysparametrejä ja painokertoimia voidaan opettaa siten, että verkko pystyy muodostamaan syötevektorista halutun vastevektorin. Se hyväksyy syötteikseen ainoastaan binäärilukuja eli nollia ja ykkösiä. Ainoa Perceptron-verkossa käytetty aktivaatiofunktio on kynnysfunktio, eli myös sen vasteet voivat olla ainoastaan ykkösiä tai nollia. Syy siihen, miksi Perceptron-verkko voi sisältää vain yhden neuronikerroksen piilee sen oppimisalgoritmissa joka ei pysty käsittelemään useampia kerroksia, eli se ei osaa välittää neuronissa laskettua vastetta toisen neuronin syötteeksi. Perceptron-verkon hyviä puolia on sen helppokäyttöisyys. Verkon huonoja puolia taas on sen ilmaisukyky eli vasteet ovat ainoastaan nollia tai ykkösiä, ja lisäksi sillä pystytään käsittelemään ainoastaan lineaarisia tapauksia.

#### Lineaarinen verkko

Lineaariset verkot poikkeavat edellä kuvatuista Perceptron-verkoista aktivaatiofunktion suhteen eli niissä hyödynnetään lineaarisia aktivaatiofunktioita. Näin ollen vasteet voivat saada arvokseen minkä tahansa lukuarvon. Näiden verkkojen avulla voidaan suorittaa esimerkiksi lineaarisia funktioapproksimaatioita. Verkot mahdollistavat myös niin sanotun Widrow-Hoff oppimisalgoritmin hyödyntämisen opetusvaiheessa. Sen toiminta perustuu lasketun vasteen ja tavoitevasteen välisestä erosta lasketun neliösummavirheen minimointiin. Tämän tyyppinen virheen minimointi toimii erittäin luotettavasti lineaarisia järjestelmiä käsiteltäessä, koska ne sisältävät ainoastaan yhden virheminin. Verkot rakentuvat edelleen ainoastaan yhdestä neuronikerroksesta eli niitä voidaan soveltaa ainoastaan lineaaristen ongelmien käsittelyyn.

#### Monikerros-Perceptron (MLP eli Multi-layer Perceptron)

Monikerros-Perceptron on edellä kuvatuista Perceptron-verkoista ja lineaarisista verkoista edelleen kehitetty neuroverkkoarkkitehtuuri. Suurin muutos yhden kerroksen Perceptroniin verrattuna on, että se muodostuu useammasta neuronikerroksesta. Tämä mahdollistaa mielivaltaiset kuvaukset syöte- ja vastevektoreiden välillä. Yhdestä sigmoid-piilokerroksesta ja lineaarisesta vastekerroksesta muodostuvan monikerros-Perceptron-verkon avulla pystytään teoriassa approksimoimaan mikä tahansa mielivaltainen funktio. Ainoa edellytys on, että piilokerroksessa on riittävä määrä neuroneita. Suurin osa tänä päivänä toteutetuista neuroverkkosovelluksista rakentuu tämän arkkitehtuurin ympärille. Monikerros-Perceptron verkkojen tärkein oppimisalgoritmi on niin sanottu Backpropagation-algoritmi ja se on saavuttanut niin suuren suosion, että sitä käsitellään yleensä jo omana menetelmänä.

#### Backpropagation(BP)-verkko

Backpropagation-verkot ovat siis Monikerros-Perceptron verkkoja, joissa hyödynnetään Backpropagation-oppimisalgoritmia, jonka mukaan nämä verkot ovat myös saaneet nimensä. Käytetty algoritmi on niin sanottu virheen takaisinsyöttöalgoritmi. Se kehitettiin



yleistämällä Widrow-Hoff oppimisalgoritmi MLP-verkoille ja epälineaarisille aktivaatiofunktioille. Epälineaarisissa tapauksissa ongelmaksi muodostuu, että virheminimejä saat-  
taa löytyä useampia. Kun opetuksessa hyödynnetään neliösummavirheen minimointia, verkko voi löytää paikallisen virheminimin jolloin oppiminen lakkaa vaikka verkko ei vielä ole läheskään ideaalinen. Lisäksi algoritmi vaatii toimiakseen erittäin pienen oppimisno-  
peus-parametrin, mikä tekee opettamisesta erittäin hitaan prosessin. Johtuen Backpropa-  
gation-algoritmin suosiosta, on sen toimintaa monin tavoin pyritty optimoimaan. Edellä  
kuvattuja ongelmia on pyritty ratkaisemaan kehittämällä algoritmia edelleen ja lisäämällä  
siihen erilaisia oppimista ohjaavia parametrejä.

Perusalgoritmin toiminta perustuu siis neuronikerrosten vasteiden virheistä lasketun neliö-  
summavirheen minimoimiseen. Tähän pyritään muuttamalla neuronien kynnysparamet-  
rien ja painokertoimien arvoja siten, että yksittäisten neuronien virhe olisi mahdollisimman  
pieni. Koska verkko muodostuu useammista kerroksista on algoritmi toteutettu siten, että  
ensin lasketaan vastekerroksen virhe, jota käytetään kyseisen kerroksen painojen korjaa-  
miseen. Tämän jälkeen lasketaan vastekerrosta edeltävän kerroksen virhe vastaavalla  
tavalla, mutta laskennassa hyödynnetään nyt vastekerrokselle laskettuja uusia arvoja.  
Näin käydään läpi verkon kaikki kerrokset kukin vuorollaan. Tästä nimi virheen takaisin-  
syöttöalgoritmi.

#### Itseorganisoituva verkko (SOM eli Self-Organizing Map)

Suomalaisen professori Teuvo Kohosen kehittämä "itseorganisoituva verkko"-algoritmi on  
tunnetuin kilpailuoppimista hyödyntävistä menetelmistä. Itseorganisoituvalla kartalla on  
kaksi tärkeää ominaisuutta: vektorikvantisointi ja jatkuva "topologinen" kuvaus kahden  
avaruuden välillä. Tämä tarkoittaa, että neuronit pyrkivät muuttamaan painokertoimiaan  
kohti tilaa, jossa jokainen neuroni on keskimäärin yhtä usein aktiivinen ja aktiivisuudet  
ovat jakautuneet mahdollisimman tasaisesti erilaisten syötteiden joukkoon. Lisäksi neuro-  
nit, jotka ovat verkossa lähellä toisiaan reagoivat samantyyppisiin syötteisiin, kun taas ver-  
kossa kauempana olevat neuronit reagoivat erityyppisiin syötteisiin. Tämän tyyppisiä  
verkkoja voidaan soveltaa esimerkiksi tehtäviin, joissa datan klusterointi ja ryhmittely ovat  
tärkeitä. /13,21,27/

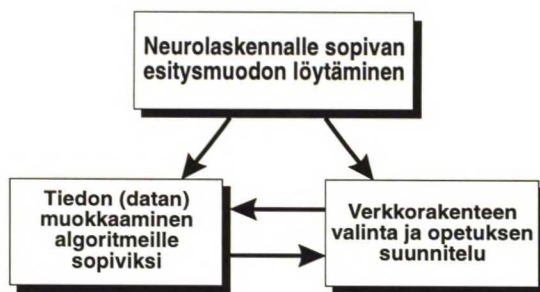
#### 4.2.4 Neurolaskennan soveltaminen käytännössä

Yksi tärkeimmistä tekijöistä neuroverkkomallinnuksen onnistumisen kannalta on ratkais-  
tavan ongelman hahmottaminen, eli mitkä ovat niitä järjestelmän piirteitä jotka ovat sen  
toiminnan kannalta merkittäviä. Tämä järjestelmää kuvaavien ominaisuuksien etsiminen  
saattaa olla kaikkein merkittävin vaihe sovelluksen onnistumisen kannalta, mutta mitään  
yleispätevää menetelmää näiden piirteiden hakemiseen ei ole pystytty kehittämään. Tässä  
vaiheessa joudutaan usein tukeutumaan järjestelmän perinpohjaiseen tutkimiseen ja ana-  
lysointiin. Seuraava vaihe on järjestelmää kuvaavien suureiden muuttaminen sellaiseen  
esitysmuotoon, että niitä pystytään neuroverkkoalgoritmeilla käsittelemään. On muistet-  
tava, että algoritmeilla pystytään käsittelemään ainoastaan numeerista informaatiota.  
Tämä käsittely jaetaan vielä yleensä tiedon esi- ja jälkikäsittelyyn. Tällä tarkoitetaan, että  
ensin tekijät muunnetaan neuroverkon ymmärtämään muotoon, minkä jälkeen neurover-  
kon vasteet on jälleen muunnettava takaisin sellaiseen esitysmuotoon, että ne ovat tulkitta-  
vissa. Kolmas vaihe on käytettävän verkkorakenteen suunnittelu ja opetusalgoritmien  
valinta siten, että verkolla saavutetaan halutut tulokset. (kuva 14) /27/

Kun järjestelmää ryhdytään mallintamaan neurolaskentaa hyväksi käyttäen on tärkeää  
ymmärtää, että neurolaskenta ei ole mikään helppo tekniikka, joka antaa pienellä vaivalla  
välittömästi tarkkoja tuloksia. On täysin tapauskohtaista, milloin neurolaskentaa ylipää-  
tään voidaan soveltaa ongelman ratkaisuun. Lisäksi on tärkeää ymmärtää neuroverkkojen  
opetusalgoritmien vaatimukset ja rajoitukset. Tässä mielessä neurolaskenta ei ole lähes-



kään niin "valmis" tekniikka kuin esimerkiksi sumea logiikka, jonka soveltaminen on varsin suoraviivaista. Merkittävimmän ongelman neuroverkkojen tapauksessa muodostaa yleensä käytetty opetusaineisto. Sitä voi olla liian vähän tai se ei kuvaa riittävällä tarkkuudella mallinnettavaa järjestelmää, jolloin siitä on mahdoton oppia mitään järkeviä toimintamalleja. Vaikka edellä kuvatut ongelmat on tiedostettu, ei ole olemassa mitään yleistä sääntöä siitä, miten paljon opetusaineistoa tarvitaan ja millaista sen tulisi olla rakenteeltaan, jotta mallinnusprosessi olisi ideaalinen. Seuraavassa on lyhyesti käyty läpi muutamia yleispäteviä sääntöjä, joita voidaan käyttää neuroverkkomalleja suunniteltaessa ja sovellettaessa. Edelleen on kuitenkin painotettava, että ohjeet ovat ainoastaan viitteellisiä, eivätkä ne ehdottoman varmasti takaa onnistunutta mallinnusta.



**Kuva 14.** Neurolaskennan soveltamisen edellytykset. /27/

Neuroverkkojen opetusmenetelmät eivät valitettavasti ole samalla tasolla kuin vakiintuneet matemaattiset menetelmät. Neurolaskennassa on yleensä pakko tyytyä "iteratiiviseen" ratkaisuun, jossa täsmälliseen ratkaisuun pyritään toistuvien yritysten kautta. Hyvä mitta opetuksen onnistumiselle on verkon kyky yleistää saatu ratkaisu. Tällä tarkoitetaan verkon kykyä opetuksen jälkeen mallintaa opetusaineistoon kuulumattomia tapauksia. Usein käy kuitenkin niin, että vaikka malli oppii opetusaineisto erittäin hyvin, mallin yleistyskyky on erittäin huono. Tämä johtuu siitä, että verkko oppii opetusaineiston esimerkit yksittäisinä tapauksina ikäänkuin ulkomuistista. Yleistyksen onnistumista kannattaa useimmiten testata uudella, opetusaineistoon kuulumattomalla datalla. /27,44/

Yksi tärkeimpiä neuroverkon oppimiskykyyn vaikuttavia tekijöitä on siis sen sisältämien neuronien lukumäärä. Mitä enemmän neuroneita verkossa on, sitä helpommin se oppii opetusaineiston. Tässä on vaarana, että jos neuroneita on liikaa, verkon yleistyskyky heikkenee eli se oppii opetusaineiston ulkoa. Jos taas neuroneita on liian vähän, ei verkko pysty yleistämään mallia riittävällä tarkkuudella. Neuronien määrän optimointiin ei ole olemassa mitään täysin luotettavaa keinoa, mutta joitakin yleisiä sääntöjä on kokemuksen pohjalta laadittu:

- Koko verkon painokertoimien lukumäärä ei saisi ylittää syötevektorin kokoa kerrottuna syötevektoreiden määrällä.
- Monikerrosverkoissa tulee piilokerrosten neuronien lukumäärän olla selvästi opetusaineiston kokoa pienempi.
- Jos tehtävä vaatii kaarevia päätöspintoja, tulee monikerrosverkoissa piilokerrosten neuronien määrän olla osapuilleen kolme kertaa syötevektorin koko (dimensio). /27/



## 5

## KOESUUNNITELMA

Diplomityön tavoitteena on useammasta tulostimesta muodostuvan hajautetun painatusjärjestelmän suorituskyvyn ennustamiseen soveltuvan mallin kehittäminen. Kirjallisuusosassa luvussa 4 tutustuttiin oppiviin ja älykkäisiin mallinnusmenetelmiin, joista parhaiten tämän tyyppisen järjestelmän mallintamiseen soveltuvaksi katsottiin neurolaskenta. Sen avulla pystytään selvittämään epälineaarisia riippuvuuksia suuresta datamäärästä ja sitä voidaan lisäksi soveltaa myös ennen kokemattomien tilanteiden ennustamiseen. Jotta neurolaskentaa voitaisiin soveltaa, tarvitaan opetusaineistoa jonka avulla mallille pystytään opettamaan järjestelmän suorituskyvyn vaikuttavat ominaisuudet ja riippuvuudet. Lisäksi tarvitaan testiaineistoa, jolla kehitettyjen mallien tarkkuutta voidaan arvioida. Seuraavassa on esitelty opetus- ja testiaineiston rakenne ja käyty läpi koeympäristön järjestelmämuuttujat niiltä osin, kun ne ovat ilman erillisiä kokeita selvitettävissä. Lopuksi on kerrottu lyhyesti suoritettavista kokeista, joiden avulla on tarkoitus selvittää suorituskymallin kehittämiseen tarvittavat suorituskyyä kuvaavat parametrit.

## 5.1 Käytetyn materiaalin rakenne

### 5.1.1 Opetusaineisto

Tavoitteena oli rakentaa testitulostuksissa käytetty materiaali mahdollisimman heterogeeniseksi eli opetusaineistossa pyrittiin varioimaan kaikkia mahdollisia muuttujia, joilla on havaittu olevan enemmän tai vähemmän vaikutusta tulostusaikoihin. Liitteessä 1 on esitetty opetusmateriaaliksi tarkoitettujen tiedostojen rakenne ja niiden sisältämien elementtien datamäärät. Kuten taulukosta voidaan havaita, on aineistoa luotaessa varioitu mahdollisimman monia muuttujia, jotta siitä saataisiin mahdollisimman kattava ja reaalisia tulostettavia tiedostoja kuvaava. Tiedostoja on rakennettu siten, että ne sisältävät ainoastaan tekstiä, viivakuvia tai sävykuvia. Tämän lisäksi on rakennettu tiedostoja, jotka sisältävät edellisiä elementtejä eritavoin yhdistettynä. Viiva- ja sävykuvien tapauksessa on lisäksi varioitu myös niiden kuvaformaatteja. Dokumenttien sivumäärää on vaihdeltu ja tekstin tapauksessa on varioitu myös yhdelle sivulle sijoitettujen fonttien määrää. Käytettyjen viiva- ja sävykuvaformaattien ominaisuuksista on lyhyesti kerrottu kirjallisuusosan kappaleessa 2.2.2 taulukoissa 1 ja 2.

Koska yksi testitulostuksissa käytetyistä tulostimista ei tue Postscript level 2:a, testimateriaalin tekstiä varioitaessa käytettiin ainoastaan level 1:sen fontteja. Näitä ovat *Courier*, *Helvetica*, *Symbol* ja *Times-Roman*. Tällä haluttiin varmistaa, että erilaisiin fonttikonversioihin kuluva aika ei vääristäisi tulostusaikoja. Koska fontteja oli käytössä näin vähän, käytettiin näistä myös erilaisia kirjasinleikkauksia eli *Italic*/Obliqueta, *Bold* ja *Bold-Italic*/Bold-Obliqueta. Fonttikokona käytettiin kaikissa tiedostoissa 12 pt. Fonttikokoa ei katsottu tarpeelliseksi varioida, koska aiemmissa tutkimuksissa on käynyt ilmi, että font-

tien variointitavalla ei ole merkittävää vaikutusta tulostusaikaan; ainoa merkitsevä tekijä on fonttivaihtojen lukumäärä sivulla.

Kaikki testimateriaali taitettiin Frame Maker 4.02 taitto-ohjelmalla. Kaikki kuvat sijoitettiin dokumentteihin linkkien avulla, joten kuvaformaattien datamäärät oli helposti selvitetävissä. Ainoa tarvittava toimenpide oli laskea kuhunkin dokumenttiin linkitettyjen kuvatiedostojen tiedostokoot yhteen. Tekstin datamäärä määritettiin tallentamalla tiedosto "vain teksti"-muodossa, jolloin muodostuneen tekstitiedoston koko kertoo suoraan absoluuttisen datamäärän. Fonttien lukumäärä huomioitiin jo tiedostojen generointivaiheessa eli sitä ei tarvinnut opetusaineistosta erikseen selvittää. Postscript-koodia generoitaessa käytettiin Apple Laserwriter II NTX-ajuria. Kyseinen ajuri valittiin sillä perusteella, että sitä suositellaan käytettäväksi esimerkiksi Xeroxin Docutech-tulostimella tulostettavaksi aiotujen Postscript-tiedostojen generointiin.

### 5.1.2 Testiaineisto

Yhden tulostimen suorituskyvyn ennustamiseen tarkoitettujen neuroverkkomallien testaamiseen käytettiin samaa testiaineistoa, jota Pertti Mäkitalo käytti diplomityössään /33/. Tämä testiaineisto muodostuu yhdeksästä tiedostosta, joissa on varioitu lähinnä sävy- ja viivakuvaformaatteja. Testimateriaalin rakenne ja sen sisältämien elementtien datamäärät on esitetty liitteessä 3 taulukossa 1. Taulukko on jaettu kolmeen osaan. Viisi ensimmäistä tiedostoa sisältävät osia TKK:n graafisen tekniikan laboratoriossa käytetystä oppikirjasta "Fundamentals of Printing Technology". Kaikki tiedostot ovat ulkoasultaan täsmälleen toisensa kaltaisia. Ainoastaan niiden sisältämien viiva- ja sävykuvien formaatteja on varioitu. Tiedostonimen ensimmäinen osa kertoo käytetyn viivakuvaformaatin ja toinen sävykuvaformaatin. Näin ollen esimerkiksi tiedoston *wmfeps.ps* viivakuvat ovat wmf-formaatissa ja sävykuvat eps-formaatissa. Kolme seuraavaa tiedosto sisältävät "Graafisen kerhon matrikelin", jossa on käytetty ainoastaan sävykuvia joiden formaatti voidaan lukea suoraan tiedoston nimestä *grak*-etuliitteen jälkeen. Viimeinen tiedosto sisältää ainoastaan tekstiä, joka on taitettu kahteen palstaan.

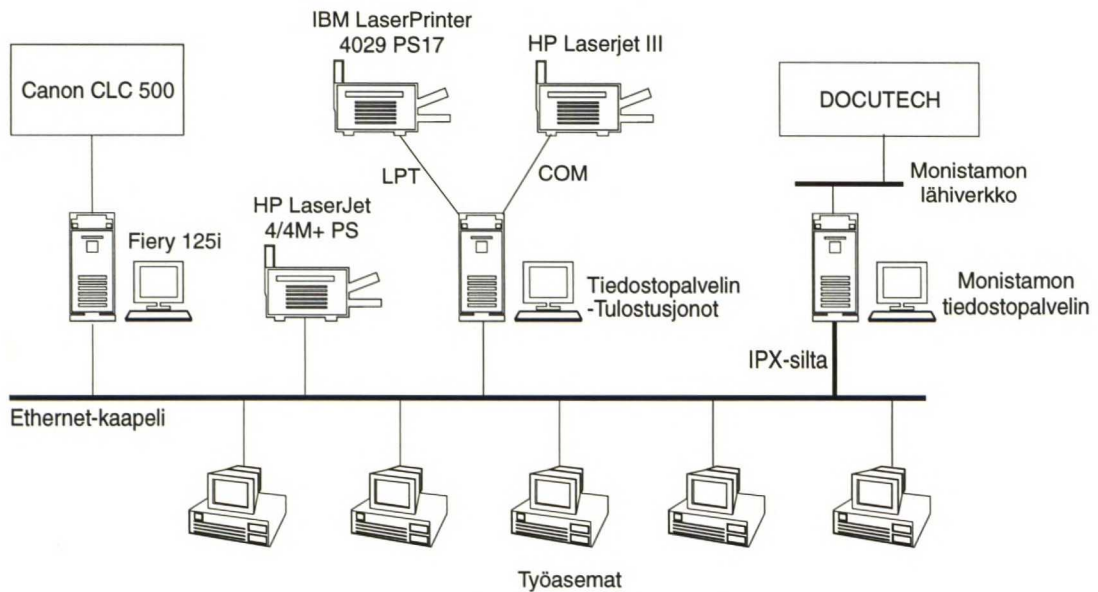
## 5.2 Koeympäristö

Kuvassa 15 on esitetty kaaviokuva Teknillisen korkeakoulun Graafisen tekniikan laboratorion lähiverkon rakenteesta, sen tulostamisen kannalta olennaisista kytkennöistä ja siihen liitetyistä laitteista. Kaikki diplomityön testitulostukset suoritettiin tässä ympäristössä. Todellisuudessa verkko muodostuu useammasta segmentistä, eli työasemat ja tulostimet eivät ole fyysisesti kytkettyinä samaan kaapeliin siten kuin kuvassa esitetään. Segmentit on kuitenkin yhdistetty ainoastaan yksinkertaisen toistimen avulla, joten yhdessä segmentissä tapahtuva liikenne vaikuttaa kaikkien muiden segmenttien kuormitukseen. Näin ollen huolimatta segmentoinnista voidaan verkkoa käsitellä ikäänkuin kaikki sen osat olisi kytketty fyysisesti samaan kaapeliin. Verkko on topologiaaltaan väylä, kaapelointi on toteutettu ohuella 10BASE2-koaksiaalikaapelilla ja siinä käytetään Ethernet-lähiverkkotekniikkaa (CSMA/CD). Verkon teoreettinen siirtokapasiteetti on 10 Mbit/s, mutta todellisuudessa sen tehollinen suorituskkyky on alle puolet tästä (noin 40%) johtuen Ethernetin käyttämästä kilpavaraustekniikasta.

Verkon hallinnasta huolehtii verkkopalvelin AST Manhattan Series V 5090, joka rakentuu yhden Pentium-prosessorin ympärille. Muistia palvelimella on 32 Mt. Diplomityön testitulostusten kannalta merkittävät palvelimen ulkoiset liitännät on toteutettu 32 bittisten PCI(Peripheral Component Interconnect)- tai EISA(Extended Industry Standard Architecture)-väylien kautta. Palvelimen kautta on käytössä 2 SCSI(Small-Computer System Inter-



face)-levyä, joista kumpikin on kooltaan 4 Gt. Todellisuudessa kerrallaan on käytössä vain 4 Gt johtuen palvelimen varmistusjärjestelmästä; kaikki tieto on tallennettu kahteen kertaan eli toiselle levyistä on luotu niin sanottu "mirror"-partitio. Fyysisesti palvelin on kytketty tavallisen työaseman tavoin yhteen toistimella yhdistetyistä segmenteistä eli liikenne palvelimen ja verkkolaitteiden välillä kuormittaa verkkoa normaalilla tavalla.



**Kuva 15.** Graafisen tekniikan laboratorion lähiverkko, sekä siihen kytketyt työasemat, tulostimet ja palvelin. Xeroxin Docutech-tulostimeen ei ole käytössä suora yhteyttä, vaan tulostettavat tiedostot lähetetään IPX-sillan kautta monistamon palvelimelle. Sieltä ne on erillisenä työvaiheena siirrettävä Docutechin tulostinpalvelimelle.

Verkkokäyttöjärjestelmä on Novell NetWare 4.1. Sen tukemat siirtoprotokollat ovat IPX(Internetnetwork Packet Exchange), TCP/IP<sup>1</sup>(Transmission Control Protocol/Internet Protocol) ja AFP(AppleTalk Filing Protocol), joka on AppleTalkin kehittyneempi versio. IPX on Novellin oma protokolla, joka mahdollistaa osoitteiden välisen pakettimuotoisen datasiirron lähiverkoissa. Osoitteet voivat sijaita samassa lähiverkossa, sekä sillan tai reitittimen kautta yhdistetyssä toisessa lähiverkossa. TCP/IP on verkkoprotokolla, joka mahdollistaa ominaisuuksiltaan poikkeavien verkkosolmujen kommunikoinnin heterogeenisessä ympäristössä. AFP on protokolla, joka tarjoaa tarvittavat kommunikointi ja datasiirtopalvelut AppleShare verkkoon kytketyille palvelimille ja laitteille.

Edellä kuvattu tiedostopalvelin, yhdessä verkkokäyttöjärjestelmän kanssa huolehtii myös verkkotulostuksen ohjauksesta ja hallinnasta eli testiympäristössä ei ole käytössä erillistä tulostinpalvelinta. Tämä lisää kuormitusta palvelimella ja sitä kautta vaikuttaa myös tulostusjärjestelmän kokonaissuorituskykyyn. Tulostimien tulostusjonot sijaitsevat fyysisesti palvelimen levyllä, jolle tulostettavat työt tallennetaan väliaikaisesti odottamaan tulostusvuoroaan ja siirtoa valitulle tulostimelle rasterointia varten. Tästä syystä on huolehdittava siitä, että verkkolevyillä on aina riittävästi tilaa varattuna ja vapaana tulostusjonoja varten, muussa tapauksessa joudutaan tilanteeseen, jossa järjestelmä ei enää pysty vastaanottamaan tulostettavia töitä eli se menee tukkoon. Myös tulostimien työmuistin koko on yleensä hyvin rajallinen eli tulostusvuorossa oleva työ joudutaan siirtämään rasterointiprosessorin käsiteltäväksi pienissä palasissa, joiden koko riippuu tulostimen työmuistin

1. Internet-verkon käyttämä protokolla.

koosta. Myös tästä syystä on tärkeää, että verkkolevyjen tallennuskapasiteetti on riittävä. Lisäksi tästä seuraa, että suuria tiedostoja tulostettaessa verkko ja palvelin kuormittuvat edelleen lisää.

Graafisen tekniikan laboratoriossa voidaan verkon kautta ilman ylimääräisiä toimenpiteitä tulostaa neljälle tulostimelle. Nämä ovat

- HP LaserJet III,
- HP LaserJet 4/4M+ PS,
- IBM LaserPrinter 4029 PS17 ja
- Canon CLC 500 + Fiery 125i RIP.

HP LaserJet III on niin vanha ja hidas tulostin, että sen hyödyntämisestä testitulostuksissa luovuttiin kokonaan. HP LaserJet 4/4M+ PS on ainoa "aito" verkkotulostin eli se on kytketty verkkoon Ethernet-verkkokortin avulla. IBM LaserPrinter 4029 PS17 ei ole varsinaisesti verkkotulostin, vaikka sitä voidaan käyttää verkon kautta. Se on kytketty tiedostopalvelimen rinnakkaisporttiin ja se kuormittaa verkkoa ainoastaan silloin kun data lähetetään palvelimelle, sen jälkeen ei verkkoa enää hyödynnetä lainkaan. Canon CLC 500:n ei varsinaisesti ole tulostin, vaan se on digitaalitekniikalla toteutettu värikopiokone. Sitä ohjaa verkkoon Ethernet-kortilla kytketty Efin valmistama Fiery 125i palvelin-RIP<sup>1</sup>. Koska Fieryn RIP on ohjelmistopohjainen, tarvitaan sitä varten oma työasema. Efi toimittaa yleensä järjestelmän kokonaisena pakettina, koska ohjelmisto asettaa tiettyjä erikoisvaatimuksia käytettävälle työasemalle.

Fiery-RIP on toteutettu kolmella prosessorilla, joista ensimmäinen (Intel 80186) huolehtii ainoastaan RIP:n I/O-toiminnoista. Näin ollen työn vastaanottaminen ei vaikuta rasterointiprosessorin muihin toimintoihin. Toinen prosessori (Intel 80386) huolehtii normaaleista käyttöjärjestelmätason toiminnoista ja kolmannella (MIPS R3000) hoidetaan varsinaisen rasteroinnin vaatima raskas "numeronmurskaus". Huolimatta siitä, että Fiery 125i on ikäänkuin erillinen tulostinpalvelin, kaikki Canonilla tulostettavat tiedostot siirretään ensin tiedostopalvelimen levyille ja sieltä edelleen Fieryn omalle levyille. Tämän jälkeen tulostaminen ei kuitenkaan enää kuormita varsinaista verkkojärjestelmää lainkaan. HP ja Fiery 125i v2.3 tukevat Postscript level 2:ta, IBM on sen verran iäkkäämpi tulostin, että se tukee ainoastaan level 1. Tulostimien tärkeimmät tekniset ominaisuudet on koottu taulukkoon 4.

Taulukko 4. Testeissä käytettyjen tulostimien tekniset tiedot.

Tulostin	Nopeus (s/min)	Muisti (Mt)	Resoluutio (dpi)	Prossessorin teho (MHz)
HP LaserJet 4/4M+ PS	12	6	600	33
IBM LaserPrinter 4029 PS17	10	5	300 (600 RET <sup>a</sup> )	16,7
Canon CLC 500 + Fiery 125i	10 –	– 64	– 400	– 36

a. Ei käytetty testitulostuksissa.

CLC 500:n manuaalissa ilmoitettu 10 s/min sivuntulostusnopeus on realistinen ainoastaan siinä tapauksessa, että samasta sivusta tulostetaan useampia kappaleita eli konetta käy-  
tään kuten kopiokonetta. Tämä johtuu siitä, että Fiery 125i pystyy käsittelemään ainoas-  
taan yhtä sivua kerrallaan. Käytännössä RIP ohjaa kopiokonetta niin kauan kunnes täysin  
valmis sivu on luovutuskaukalossa. Tänä aikana prosessori ei pysty suorittamaan mitään

1. Graafisen tekniikan laboratoriossa on ohjelmistosta käytössä versio 2.3.



muita toimintoja. Valmiiksi rasteroidun sivun siirtämiseen paperille kuluu aina vakiolliset 20 sekuntia riippumatta sivun sisällöstä eli käytännössä koneistonopeudeksi muodostuu 3 sivua/minuutissa. HP:n ja IBM:n tapauksessa tilanne on huomattavasti parempi. Paperin läpimenoaika koneistossa on Canon CLC 500:aa nopeampi johtuen siitä, että paperin on kuljettava fyysisesti lyhyempi matka. Lisäksi rasterointiprosessorin ei tarvitse ohjata tulostimen koneistoa sen jälkeen kun sivu on valmis, vaan se voi aloittaa välittömästi uuden sivun rasteroinnin. Koneisto hoitaa automaattisesti ilman prosessorin ohjausta paperin kuljettamisen fotojohteelta luovutuskaualoon.

Xerox Docutech tulostinta ei testitulostuksissa käytetty, koska se ei ole suoraan kytketty laboratorion verkkoon, vaan yhteys on toteutettu IPX-sillan kautta. Lisäksi sen kautta saadaan yhteys ainoastaan monistamon palvelimeen, eli laboratorion lähiverkkoon kytketyiltä työasemilta ei prosessin missään vaiheessa ole suoraa yhteyttä Docutech-tulostimeen. Tämä tarkoittaa käytännössä, että ensin on erillisenä työvaiheena muodostettava yhteys palvelimelle. Tämän jälkeen tulostettavat Postscript-tiedostot kopioidaan monistamon tiedostopalvelimen levyille, josta ne edelleen siirretään Docutechin ohjausyksikön käsiteltäväksi tulostimen käyttöhenkilöstön toimesta. Koska kysymyksessä ei ole on-line kytkentä, järjestelmä sisältää runsaasti ennustamattomia tekijöitä, mikä tekee tulostimen suorituskyvyn matemaattisen mallintamisen hankalaksi.

### 5.3 Suoritettavien kokeiden kuvaus

Kun diplomityön tavoitteeksi asetettua "hajautetun painatusjärjestelmän"-mallia ryhdyttiin muodostamaan, oli ensimmäinen tehtävä käytettävän mallinnustekniikan valinta. Aiemmissa tutkimuksissa on jo todettu, että perinteisten menetelmien kuten tilastollisten analyysien ja fysiikkaan perustuvien mallien avulla ei pystytä elektronisen painatusjärjestelmän suorituskykyä ennustamaan riittävällä tarkkuudella. Tästä syystä tämän diplomityön perustavoitteena on mallintamiseen soveltuvan tekniikan etsiminen, sen testaaminen käytännössä ja hyödyntämismahdollisuuksien kartoitus. Sopiviksi tämän tyyppisten järjestelmien mallintamiseen ovat osoittautuneet niin sanotut oppivat ja älykkäät järjestelmät, joita on viime aikoina yhä enemmän ryhdytty hyödyntämään mitä erilaisimmissa sovelluksissa.

Mallinnustekniikkaa valittaessa erilaisten testien tekeminen osoittautui liian työlääksi ja aikaa vieväksi, koska jokainen tekniikoista vaatii hieman erilaisen lähestymistavan. Lisäksi mallinnettavan järjestelmän materiaali- ja systeemimuuttujia jouduttaisiin muokkaamaan kullekin menetelmälle sopivaan muotoon. Valinta tehtiin tästä syystä tekniikoita käsittelevän kirjallisen materiaalin perusteella. Suurta apua oli myös reaali maailman sovellusten kuvauksista, joita löytyy runsaasti alan kirjallisuudesta ja artikkeleista. Oppivia ja älykkäitä järjestelmiä käsittelevän kirjallisuuden perusteella tultiin siihen tulokseen, että elektronisen painatusjärjestelmän mallintamiseen parhaiten soveltuva tekniikka on ehdottomasti neurolaskenta.

Kävi kuitenkin ilmi, että neurolaskennan soveltaminen suoraan kokonaisen painatusjärjestelmän mallintamiseen on kömpelö ja epäkäytännöllinen lähestymistapa. Tästä syystä päädyttiin ennustamaan hajautettujen painatusjärjestelmien suorituskykyä siihen kuuluvien yksittäisten tulostimien suorituskyvyn kautta. Tällöin tulee ensin pystyä ennustamaan järjestelmään kuuluvan yksittäisen tulostimen suorituskyky riittävällä tarkkuudella. Jollei yksittäisten tulostimien mallintaminen onnistu, laajemmassa mallissa epätarkkuudet kumuloituvat lopputulokseen ja ennuste ei ole kovinkaan luotettava. Testien ensimmäisessä vaiheessa tulostettiin kaikki liitteessä 1 esitetyt Postscript-tiedostot jokaisella testitulostimeksi valituista kolmesta tulostimesta. Koska tavoitteena on ennustaa tietyn tulostimen kokonaissuorituskykyä, ei datasiirtoa, rasterointia ja varsinaista tulostusta työvaiheina millään tavoin eroteltu; ainoa mitattu suure on näin ollen kokonaistulostusaika. Tulostuk-



sen katsottiin alkavan sillä hetkellä, kun tiedostoa ryhdyttiin siirtämään työasemalta ja päättyvän sillä hetkellä, kun viimeinen sivu on tulostimen luovutuskaukalossa.

Varsinainen tulostusoperaatio haluttiin pitää mahdollisimman yksinkertaisena ja tehokkaana. Parhaaksi vaihtoehdoksi osoittautui Postscript-tiedoston kopioiminen haluttuun tulostinporttiin suoraan DOS-käyttöjärjestelmän komentoriviltä. Komento on tällöin muotoa:

```
copy tst025.ps lpt1
```

*Copy* on normaali DOS-käyttöjärjestelmän kopiointikomento, *tst025.ps* on esimerkki tulostettavan tiedoston nimestä ja *lpt1* tulostinportti, johon tulostettava tiedosto halutaan ohjata. Sinänsä Postscript-tiedoston lähettämistavalla ei mallin ennustustarkkuuden kannalta ole merkitystä, kunhan tulostustapa opetusvaiheessa vastaa todellisuutta eli tapaa jolla Postscript-tiedostot normaalisti lähetetään tulostimille. Diplomityössä päädyttiin komentoriviltä kopiointiin käytännön syistä, eli se oli nopein Postscript-tiedostojen lähettämiseen tarjolla olleista vaihtoehdoista.

Saatujen tuloksien perusteella pyrittiin erottamaan tulostettavasta materiaalista ne piirteet, jotka vaikuttavat tulostusaikoihin kaikkein eniten. Analyysissä hyödynnettiin muun muassa tilastollisia menetelmiä, kuten lineaarista regressioanalyysiä. Regressiomalleja käytettiin kuitenkin ainoastaan analysointivaiheessa, koska jo aiemmin on havaittu, että niiden avulla ei riittävällä tarkkuudella pystytä tulostusaikoja ennustamaan. Tulostusjärjestelmän teknisiin ominaisuuksiin ei tässä vaiheessa kiinnitetty huomiota, vaan keskityttiin etsimään ainoastaan suorituskyykyyn vaikuttavia materiaaliominaisuuksia. Tämä on perusteltua siitä syystä, että aiempien kokemusten perusteella nimenomaan ne aiheuttavat eniten epätarkkuutta tulostusaikoja ennustettaessa.

Analyysivaiheessa havaittuja, selvästi tulostusaikoihin vaikuttavia materiaaliominaisuuksia käytettiin syöteparametreina neuroverkkomallia opetettaessa ja testattaessa. Järjestelmän kuvaamiseen soveltuvaa neuroverkkomallia kehitettäessä tutkittiin muun muassa erilaisten toteutusarkkitehtuurien, opetusalgoritmien ja verkon neuronien määrän vaikutusta ennustustarkkuuteen. Näitä muuttujia varioimalla pyrittiin löytämään ideaalinen verkkorakenne, jonka ennustustarkkuus olisi riittävä. Neuroverkkoja opetettaessa käytettiin myös aiempien tutkimuksien perusteella merkittäviksi havaittuja tulostettavan materiaalin ominaisuuksia opetusparametreinä. Tällä tavoin oli tarkoitus tutkia, mille tasolle tulostettavan materiaalin kuvaus on viety, jotta järjestelmän suorituskyykyä voitaisiin neuroverkkomallilla ennustaa riittävällä tarkkuudella.

Hajautettujen järjestelmien suorituskyykyyn ennustamiseen soveltuvien mallien kehittämisessä luovuttiin neurolaskennan soveltamisesta. Tämä on perusteltua kahdestakin syystä. Ensinnäkin erilaisia tulostusarkkitehtuureja voidaan suppeassakin ympäristössä muodostaa runsaasti ja neurolaskentaa hyödynnettäessä näistä jokainen olisi mallinnettava omana erillisenä kokonaisuutenaan. Toiseksi edellä kuvattujen testitulostusten yhteydessä havaittiin, että järjestelmän ominaisuuksien vaikutukset tulostusaikoihin ovat riittävällä tarkkuudella lineaarisia. Tilanne muuttui neuroverkkomallinnukseen verrattuna siten, että algoritmeissa oli otettava huomioon myös järjestelmämuuttujia, kun ne neuroverkkomalleissa huomioitiin osana kokonaisuutta. Tämä ei sinänsä muodostunut ongelmaksi, koska suorituskyykyyn oleellisesti vaikuttavat järjestelmämuuttujat olivat helposti selvitettävissä testitulostuksien perusteella. Hajautettuja järjestelmäarkkitehtuureja simuloitiin periaatteessa kolmella tavalla:

1. Tulostamalla yhdeltä työasemalta useampia tiedostoja peräkkäin samalle tulostimelle. Tätä voidaan pitää eräänlaisena "pipeline"-simulaationa.
2. Tulostamalla yhdeltä työasemalta usealle tulostimelle.
3. Tulostamalla usealta työasemalta usealle tulostimelle samanaikaisesti.



## 6

# NEUROLASKENNAN SOVELTAMINEN MALLINNUKSESSA

## 6.1 Opetusmateriaalin analysointi

Tässä kappaleessa on tavoitteena selvittää neuroverkkomallin opetukseen tarkoitetun materiaalin sisältämien elementtien vaikutusta tulostusaikoihin. Kysymyksen voisi muotoilla esimerkiksi seuraavalla tavalla: Mitkä materiaalin ominaisuudet ovat niin merkittäviä, että ne tulisi huomioida kehitettävissä neuroverkkomallissa syöteinä? Kääntäen voitaisiin sanoa, että on pyritty selvittämään mitkä tulostettavan materiaalin ominaisuudet aiheuttavat järjestelmässä epälineaarisuutta. Aiemmissa tutkimuksissa on jo todettu, että riippuvuudet eivät ole lineaarisia. Koska testitulostukset tehtiin kolmella ominaisuuksiltaan huomattavasti toisistaan poikkeavalla tulostimella, myös analyysit on eroteltu erillisiksi kokonaisuuksiksi. Tällä on pyritty siihen, että tulostimien ja materiaaliominaisuuksien yhteisvaikutukset tulostusaikoihin saataisiin eriteltyä mahdollisimman tarkasti.

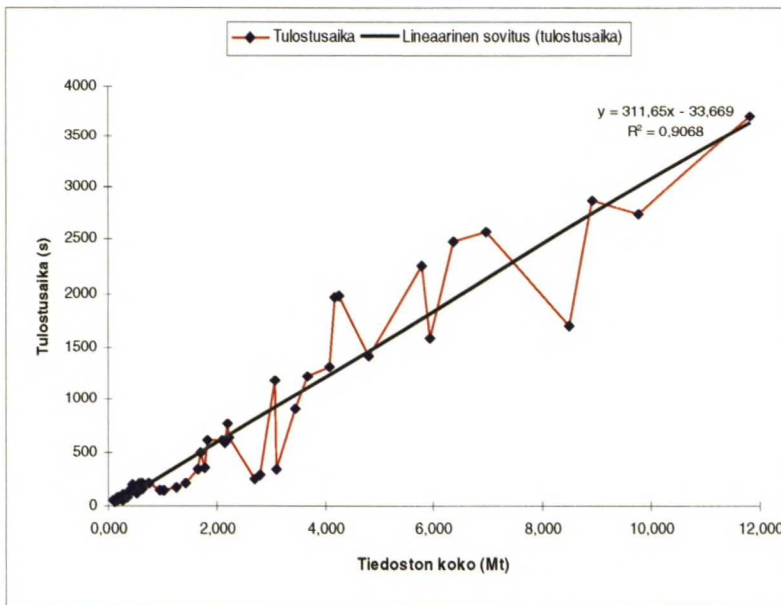
Pertti Mäkitalo pyrki diplomityössään /33/ elementtipohjaisen mallin lisäksi ennustamaan tulostusaikoja lineaarisen regressioanalyysin avulla; kehitettyjen mallien tarkkuus oli kuitenkin heikko. Regressioanalyysissä hän käytti selittävinä suureina tiedoston kokoa, sekä sivumäärää. Näistä tiedoston koko korreloi tulostusajan kanssa huomattavasti paremmin kuin sivumäärä. Hän havaitsi lisäksi, että sekä elementtipohjaisen mallin, että lineaarisen regressiomallin virheet olivat suunnilleen samaa suuruusluokkaa. Tästä syystä voitaneen tehdä oletus, että epätarkkuuden lähde on kummassakin tapauksessa sama. Neuroverkon opetusmateriaalin tulostusaikoja analysoitaessa käytetään myös tässä työssä lineaarista regressioanalyysiä. Tavoitteena on pystyä erottelemaan aineistosta ne mittauspisteet, jotka aiheuttavat suurimman poikkeaman eli eivät käyttäydy lineaarisesti; mallien avulla ei pyritä ennustamaan järjestelmän suorituskykyä. Poikkeavia mittauspisteitä vastaavista tiedostoista on etsitty yhteistä nimittäjää, joka voisi selittää poikkeaman.

Liitteessä 4 on esitetty neuroverkkojen opetukseen tarkoitetun materiaalin tulostusajat ("yhteensä" sarake), joiden perusteella tämän luvun kuvaajat on piirretty. Mittapisteet on merkitty kuvaajiin vinoneliöillä, jotka on yhdistetty toisiinsa viivalla trendien havaitsemisen ja seuraamisen helpottamiseksi. Samassa yhteydessä on mittauspisteille laskettu regressioanalyysin avulla lineaarinen sovitus. Kuvaajista on kuitenkin vaikeaa todeta, mitkä mittauspisteet poikkeavat eniten lineaarisesta regressiomallista, koska tasavälisellä x-akselilla suurin osa niistä sijoittuu x-akselin ensimmäiselle kolmannekselle. Havainnoinnin helpottamiseksi on jokaisen tulostimen kohdalla esitetty myös tiedoston koko/tulostusaika-kuvaaja koordinaatistossa, jonka x-akseli on logaritminen. Myös näihin kuvaajiin on liitetty trendikäyrä (polynomisovitus), jotta poikkeamat olisi helpompi havaita. Kuvaajien yhteydessä on esitetty niin sanottu regressiomallin selitysaste ( $R^2$ ), joka kuvaa sitä, kuinka suuri osuus tulostusajan vaihtelusta voidaan kuvata tiedoston koon perusteella.

Mitä lähempänä  $R^2$ -arvo on 1:stä, sitä luotettavammin mallilla voidaan ennustaa tulostusaikaa kun tiedoston koko tunnetaan. Lähellä 0:aa olevat arvot puolestaan viittaavat siihen, että tiedoston koon tunteminen ei juurikaan vähennä tulostusajan ennustamiseen liittyvää epävarmuutta. Selitysasteet ovat lasketuilla lineaarisilla malleilla todella hyviä (huonoimmillaan 0,85 ja parhaimmillaan 0,91). Lisäksi tilastollisen luotettavuuden testaaminen esimerkiksi F-testillä antaa sellaisen kuvan, että saadut tulokset olisivat tilastollisesti erittäin luotettavia. Todellisuudessa mallien tekemät virheet ovat suuria ja regressioanalyysin käyttöön tässä tapauksessa tulee suhtautua hyvin kriittisesti. Tehtyjä regressiosovituksia on tulkittu ainoastaan viitteellisinä, eikä laskettuja malleja tule missään tapauksessa suoraan soveltaa suorituskkyä ennustettaessa. Tästä syystä tehtyjen tilastollisten luotettavuustestien tuloksia ei katsottu tarpeelliseksi esittää tässä yhteydessä.

### HP:n tulostusaikojen analysointi

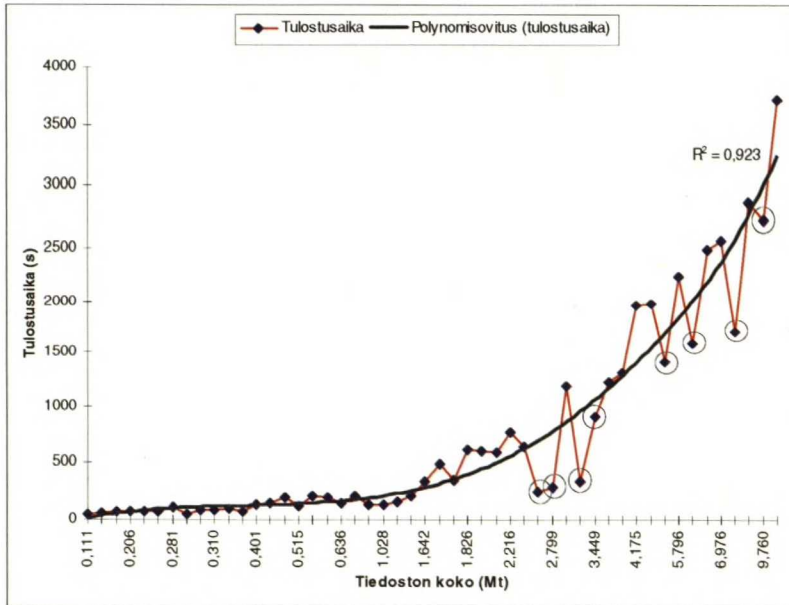
Kuvassa 16 on esitetty HP LaserJet 4/4M+ PS tulostimen tulostusajat suhteessa tulostettujen Postscript-tiedostojen kokoon. Kuten kuvaajasta havaitaan, on hajonta suuri verrattuna lineaarisen mallin ennustamaan suoraan verrattuna. Selitysaste on kuitenkin niin korkea kuin 0,91. Tämä voi johtua osittain siitä, että mittauspisteet vaihtelevat suhteellisen tasaisesti suoran kummallakin puolella. Tästä syystä nouseva suora kuvaa tapahtuvaa kokonaisvaltaista muutosta suhteellisen hyvin.



**Kuva 16.** HP LaserJet 4/4M+ PS:n tulostusajat suhteessa tulostettavan Postscript-tiedoston kokoon. Kuvaajassa on esitetty kaikki mitatut tulostusajat.

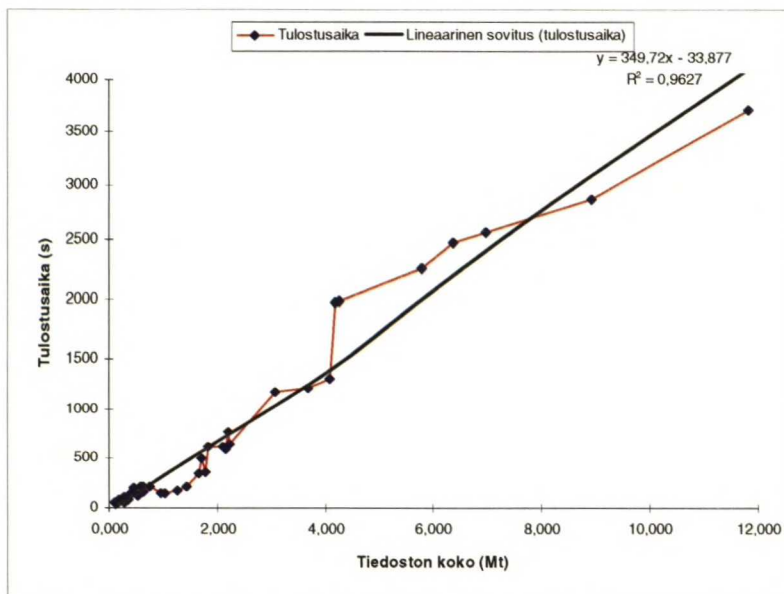
Kuvaajasta on kuitenkin vaikea havaita selviäkään poikkeamia ja tästä syystä kuvassa 17 on esitetty täsmälleen samat mittaustulokset siten, että x-akseli on logaritminen. Tästä kuvaajasta on helposti erotettavissa trendistä merkittävästi alaspäin poikkeavat mittapisteet. Ne on merkitty kuvaajaan ympyrällä. Pisteitä vastaavat opetusmateriaalin tiedostot ovat: tst013.ps, tst046.ps, tst048.ps, tst045.ps, tst033.ps, tst016.ps, tst036.ps ja tst006.ps. Järjestys on sama kuin diagrammissa ympyröillä merkittyjen pisteiden esiintymisjärjestys vasemmalta oikealle. Jos tarkastellaan tiedostojen sisältöä liitteessä 2 esitetyn taulukon pohjalta havaitaan, että tiedostot ovat ensinnäkin suhteellisen suuria eli 2,7 Mt:sta aina lähes 10 Mt:uun. Tämä ei kuitenkaan vielä riitä selittämään poikkeamaa, vaan merkittävämpi havainto on, että ne kaikki sisältävät runsaasti EPS-sävykuvia (41-100% datamäärästä).





**Kuva 17.** HP:n tulostusajat esitettynä logaritmisella x-akselilla. Nousevasta käyrästä selvästi alaspäin poikkeavat pisteet on merkitty ympyröillä (○).

Opetusaineistosta löytyy kuitenkin muutama tiedosto (tst005.ps, tst020.ps, tst025 ja tst037.ps), joiden datamäärästä vähintään 40% on EPS-sävykuvia, mutta niiden tulostusaika ei poikkea merkittävästi trendistä. Koska tiedostoja on ainoastaan neljä ja niiden väliltä ei löydy selvää yhteistä nimittäjää, on niiden osalta tyydyttävä vaikuttavia tekijöitä vain arvaamaan. Yksi syy ainakin kahden tiedoston kohdalla voisi olla niiden koko (alle 1 Mt) eli huolimatta siitä, että EPS-kuvien suhteellinen osuus datamäärästä on suuri, reaalisesti se on kuitenkin vähäinen. Näiden tiedostojen perusteella voidaan kuitenkin päätellä, että EPS-sävykuvien osuus datamäärästä ei välttämättä ole ainoa epälineaarisuutta selittävä tekijä.

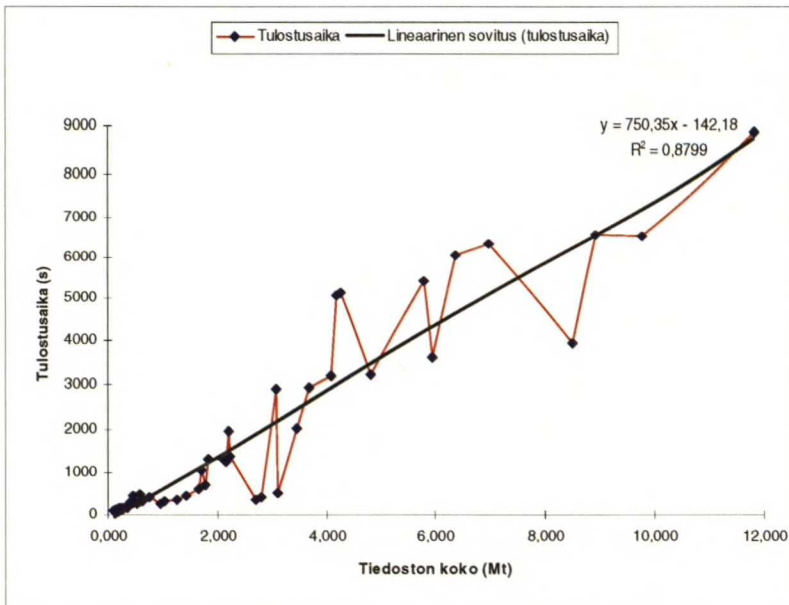


**Kuva 18.** HP:n tulokset kun kahdeksan EPS-kuvia sisältävää tiedostoa on poistettu näyteaineistosta.

Kuvasta 18 havaitaan, että edellä mainittujen kahdeksan EPS-sävykuvia runsaasti sisältävien tiedostojen poistaminen opetusaineistosta korjasi tuloksia merkittävästi lähemmäksi lineaarista mallia. Myös selitysaste parani 0,91:stä 0,96:een. Tämä ei välttämättä tässä nimenomaisessa tapauksessa olisi riittävä peruste tukemaan oletusta EPS-sävykuvien vaikutuksesta, mutta muutos parempaan on todettavissa myös visuaalisesti. Tällä perusteella niiden vaikutus voidaan luokitella jopa aika merkittäväksi. Kuvaajasta nähdään kuitenkin, että esimerkiksi alle 2 Mt:n kokoisilla tiedostoilla esiintyy lineaariseen malliin verrattuna suurehkoja poikkeamia (suhteessa kokonaistulostusaikaan), joita ei pystytä selittämään tiedostojen sisällön perusteella.

### IBM:n tulostusaikojen analysointi

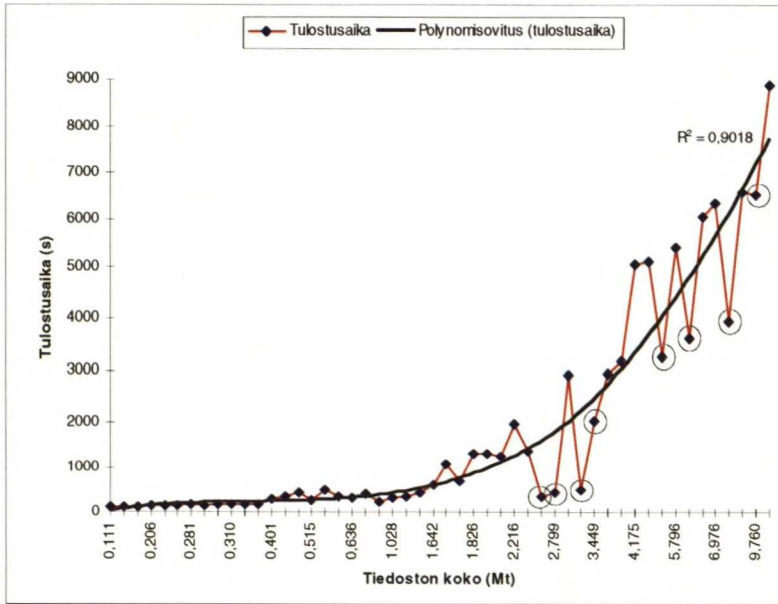
IBM LaserPrinter 4029 PS17 on perusrakenteeltaan ja toteutukseltaan hyvin samankaltainen tulostin kuin HP LaserJet 4/4M+ PS. Tämä on havaittavissa kuvassa 19 esitetyssä opetusmateriaalin tulostusaikakuvaajasta. Diagrammi on lähes identtinen HP:n vastaavan kuvaajan kanssa, joka on esitetty kuvassa 16. Huomiota kannattaa kuitenkin kiinnittää tulostusaika-akseliin eli HP:n tapauksessa tulostusajat vaihtelevat 54:n ja 3714:n sekunnin välillä, kun ne IBM:n tapauksessa vaihtelevat 104:n ja 8876:n sekunnin välillä. Tämä on tietenkin helposti selitettävissä taulukossa 4 esitettyjen suorituskkyerojen avulla. Selitysaste on hieman heikompi IBM:llä (0,88) kuin HP:llä (0,91). Tämä johtuu siitä, että vaikka kuvaajat ovat ulkoisesti lähes identtiset, on suhteellisissa tulostusajoissa hieman eroja eli IBM:n arvot poikkeavat lineaarisesta mallista hieman enemmän kuin HP:n.



**Kuva 19.** IBM LaserPrinter 4029 PS17:n tulostusajat suhteessa tulostettavan Postscript-tiedoston kokoon. Aineisto sisältää kaikki mitatut tulostusajat.

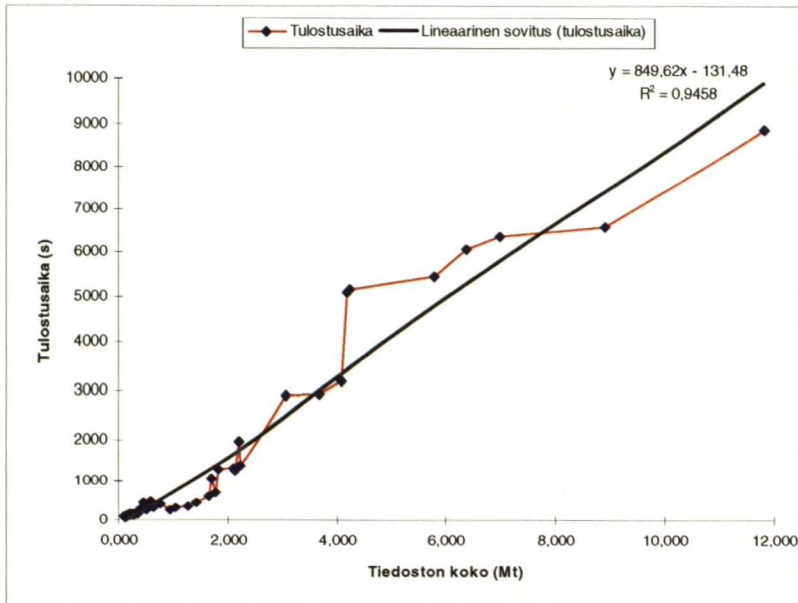
Vastaavalla tavalla kuin HP:n tapauksessa IBM:n tulostusajat skaalattiin logaritmiselle x-akselille, jolloin poikkeamat voidaan havaita huomattavasti helpommin. Myös logaritmisella asteikolla kuvaaja on lähes identtinen kuvassa 17 esitetyn HP:n vastaavan kuvaajan kanssa. Lisäksi todella selvästi trendistä alaspäin poikkeavat pisteet ovat täsmälleen samat kuin HP:llä ja niitä vastaavat tiedostot ovat myös samat. Tästä voidaan päätellä, että EPS-sävykuvien suhteellisella osuudella todellakin on epälineaarisuutta lisäävä vaikutus tulostusaikoihin, eikä kyseinen ilmiö johtunut pelkästään esimerkiksi HP:n rasterointiprosessorin ominaisuuksista.





**Kuva 20.** IBM:n tulostusajat logaritmisella asteikolla. Selvästi trendistä alaspäin poikkeavat tiedostot on merkitty vastaavalla tavalla kuin edellä eli ympyrällä (○).

Kuvassa 21 on esitetty IBM:n tulostusajat suhteessa Postscript-tiedoston kokoon EPS-sävykuvia sisältävien tiedostojen tulostusaikojen poistamisen jälkeen. Kuten HP:n tapauksessa, myös IBM:n tulostusaikadiagrammi lähestyy lineaarista mallia ja mallin selitysaste paranee huomattavasti (0,88:sta 0,95:een). Itse asiassa se parani jopa enemmän tiedostojen poistamisen jälkeen kuin HP:llä, eli voidaan olettaa että poikkeamat suhteellisissa tulostusajoissa HP:n ja IBM:n välillä johtuivat juuri poistetuista tiedostoista.

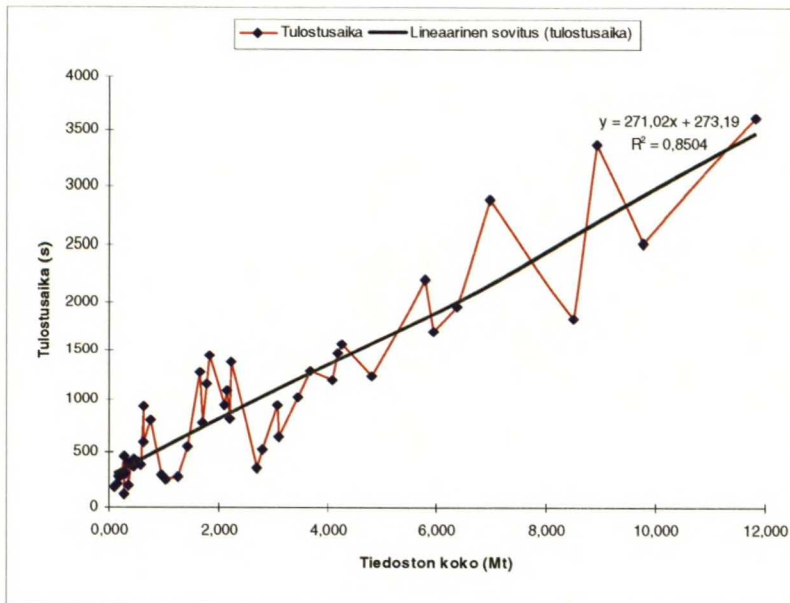


**Kuva 21.** IBM:n tulostusajat kahdeksan EPS-sävykuvia sisältävän tiedoston poistamisen jälkeen.

Myös IBM:n tapauksessa on havaittavissa sama suhteessa suurehko poikkeama lineaarisesta mallista alle 2 Mt kokoisilla tiedostoilla. Kuten jo aiemmin mainittiin, kyseisistä tiedostoista ei ainakaan tässä työssä käytetyn materiaalin perusteella ollut löydettävissä mitään yhteistä nimittäjää. Ainoa selittävä tekijä lienee tiedostojen sisältämien elementtien yhteisvaikutus eli EPS-sävykuvat eivät yksistään selitä vaihtelua, vaikka ovatkin sen merkittävä lähde.

### Canon CLC 500:n ja Fiery-RIPin tulostusaikojen analysointi

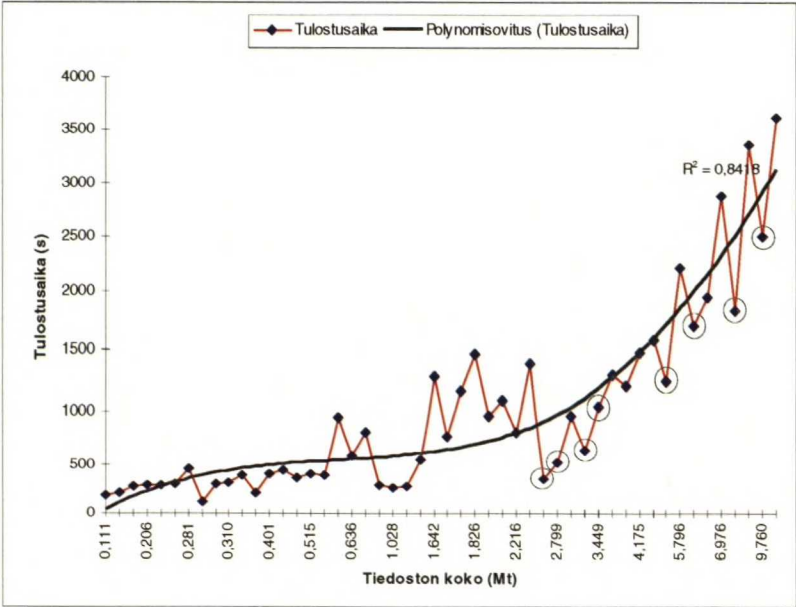
Kuvassa 22 näkyvät Canon CLC 500:n tulostusajat tasavälisellä x-akselilla. Verrattaessa kuvaajaa kahden aiemmin käsitellyn tulostimen vastaaviin diagrammeihin (kuvat 16 ja 19) havaitaan, että vaihtelu on huomattavasti suurempaa eli mitatut tulostusajat poikkeavat lineaarisesta mallista keskimäärin enemmän. Tähän viittaa myös lineaarisen mallin selityssaste (0,85), mikä sinällään ei ole kovinkaan huono. On kuitenkin pidettävä mielessä, että analysoitujen näytesarjojen tapauksessa regressioanalyysin käyttö ei ole täysin perusteltua eli selityssastetta ei voida pitää luotettavana mallin hyvyyden mittana. Kuvaajasta voidaan kuitenkin päätellä, että Canon CLC 500:n tapauksessa tulostusaikoihin vaikuttavat tekijät poikkeavat aikaisempien tulostimien yhteydessä esitetyistä, tai ainakin vaikuttavia tekijöitä on useampia.



**Kuva 22.** Canon CLC 500+Fiery 125i:n tulostusajat esitettynä tasavälisellä x-akselilla.

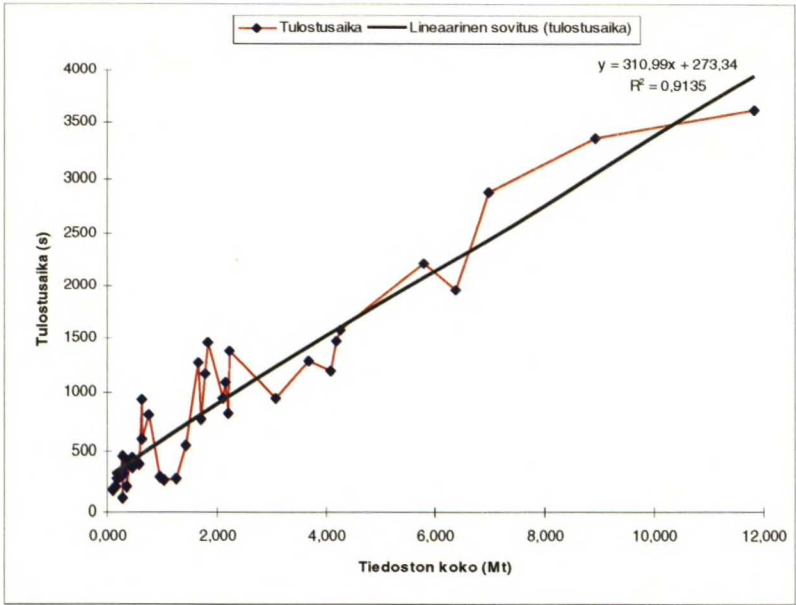
Kuten aiempien tulostimien tapauksessa, myös Canon CLC 500:n tulokset siirrettiin koordinaatistoon, jonka x-akseli on logaritminen (kuva 23). Kuten kuvaajasta havaitaan, eivät tulostusajat seuraa nousevaa käyrää läheskään samalla tarkkuudella kuin HP:n ja IBM:n tapauksessa. Selviä poikkeamia on havaittavissa erityisesti trendikäyrästä ylöspäin. Kahdella aiemmalla tulostimella ei ollut havaittavissa vastaavia piirteitä, eli on hyvin todennäköistä että kyseiset poikkeamat selittävät suuren osan mallin epätarkkuudesta. Edelleen on erotettavissa myös EPS-sävykuvia sisältävien tiedostojen aiheuttamat poikkeamat trendistä alaspäin. Nämä eivät kuitenkaan ole enää niin selviä, koska mittaustulokset sisältävät myös muita trendikäyrää selvästi vääristäviä mittapisteitä. Koska on kuitenkin oletettavaa, että näillä tiedostoilla on vastaava vaikutus kuin aiempien tulostimien tapauksessa, ne poistettiin näytesarjasta ensiksi. Nämä EPS-sävykuvia sisältävät tiedostot on merkitty kuvaajaan ympyröillä kuten aiemmissa tapauksissa.





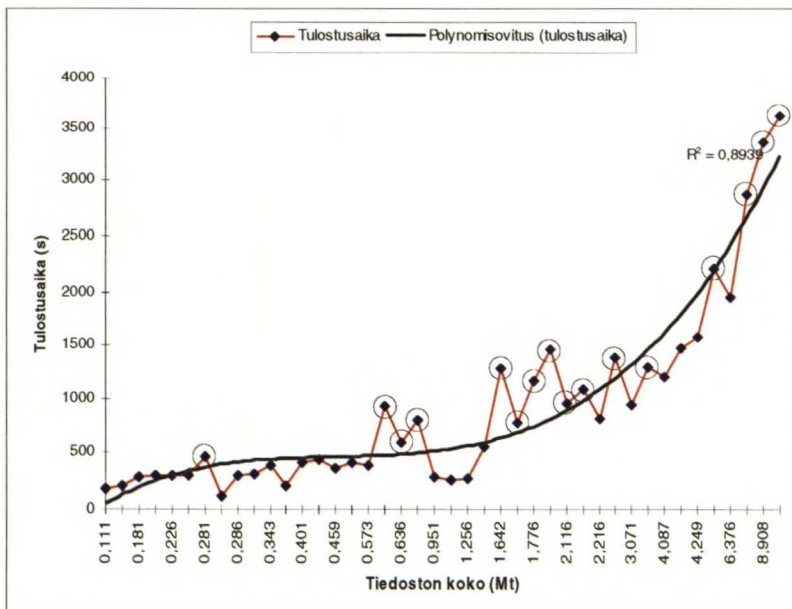
**Kuva 23.** Canon CLC 500:n tulostusajat esitettynä logaritmisella asteikolla. EPS-sävykuvia sisältävät tiedostot on merkitty ympyrällä(○).

Kuvassa 24 on esitetty mittaustulokset EPS-sävykuvatiedostojen poistamisen jälkeen. Selvästi on havaittavissa, että vaikka tilanne hieman korjaantui, on vaihtelu edelleen suurta. Vaikka selitysaste parani merkittävästi, se on edelleen huonompi kuin edellisten tulostimien tapauksessa kyseisten tiedostojen poistamisen jälkeen. Vaihtelu on kaikkein selvintä tiedostoilla, jotka ovat kooltaan alle 5 Mt. Juuri tällä alueella on erittäin vaikeaa erottaa mittauspisteitä toisistaan johtuen niiden runsaasta määrästä, joten jäljellä olevat 42 mittauspistettä piirrettiin uudelleen koordinaatistoon, jossa x-akseli on logaritminen. Tämä diagrammi on esitetty kuvassa 25.



**Kuva 24.** Tulostusajat EPS-sävykuvia sisältäviä tiedostoja vastaavien tulostusaikojen poistamisen jälkeen.

Kuvan 25 diagrammista nähdään, että vaihtelu todella on merkittävää trendikäyrän kummallakin puolella. Voidaan kuitenkin tehdä oletus, että nimenomaan käyrän yläpuolella olevat pisteet vääristävät trendikäyrää enemmän, eli ilman kyseisiä pisteitä se kulkisi käyrän alapuolella olevien pisteiden kautta. Kuvaajassa on merkitty ympyrällä nämä käyrän yläpuolella olevat, sekä hyvin lähellä sen alapuolella ylöspäin poikkeavat pisteet. Merkityt 16 tiedostoa ovat: tst041.ps, tst011.ps, tst034.ps, tst001.ps, tst044.ps, tst009.ps, tst014.ps, tst039.ps, tst049.ps, tst019.ps, tst029.ps, tst028.ps, tst008.ps, tst005.ps, tst015.ps ja tst025.ps. Liitteen 1 taulukosta on selvästi havaittavissa tiedostojen yhteinen nimittäjä eli kaikki ovat vähintään viisitoista sivuisia. Tällä perusteella voidaan suhteellisen turvallisesti tehdä johtopäätös, että sivumäärällä on Canon CLC 500:n tapauksessa merkittävä vaikutus muodostuviin tulostusaikoihin. Näin ollen kooltaan pienenkin, mutta runsaasti sivuja sisältävän tekstitiedoston tulostaminen kestää suhteettoman kauan verrattuna tiedoston kokoon.

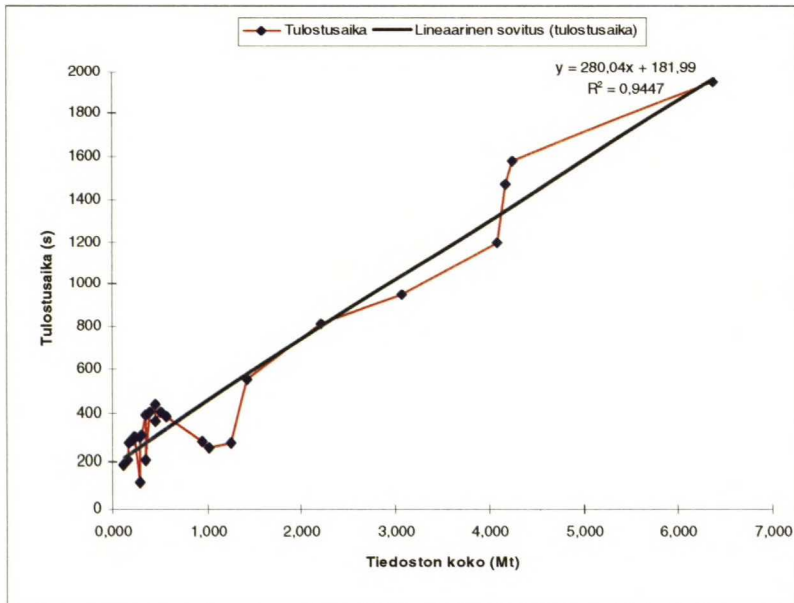


**Kuva 25.** Tulostusajat logaritmisella asteikolla sen jälkeen, kun EPS-sävykuvia sisältäviä tiedostoja vastaavat mittauspisteet on poistettu.

Tarkasteltaessa vielä liitteen 1 taulukkoa havaitaan kuitenkin, että opetusmateriaalista löytyy tiedostoja, jotka ovat vähintään 15 sivuisia, mutta eivät aiheuta selvää poikkeamaa trendistä ylöspäin. Tällaisia tiedostoja ovat: tst006.ps, tst016.ps, tst024.ps ja tst036.ps. Kolme tiedostoista kuuluu ryhmään, joka poistettiin jo ensimmäisessä vaiheessa eli ne sisälsivät EPS-sävykuvia niin paljon, että aiheuttivat itse asiassa poikkeamaa trendistä alaspäin. Neljäs eli tst024.ps sisältää EPS-viivakuvia 98%. Näin vähäisestä datamäärästä ei kovin luotettavia johtopäätöksiä voida tehdä, mutta kysymys näyttäisi olevan siitä, että EPS-sävykuvien ja sivumäärän tulostusaikoja vääristävät vaikutukset ikäänkuin kompensoivat toisiaan. Fiery 125i:n tehokas RIP rasteroi EPS-kuvat muita kuvaformaatteja nopeammin, mutta Canonin koneistonopeus puolestaan hidastaa tulostamista.

Kuvassa 26 on esitetty tulostusajat tiedostokoon funktiona sen jälkeen kun edellä merkityt runsaasti sivuja sisältävät tiedostot on poistettu testiaineistosta. Kuten havaitaan selityksaste paranee 0,91:stä 0,94:ään ja tapahtunut parannus on havaittavissa myös visuaalisesti. Kuitenkin on huomattava, että kuten aiemmissa tapauksissa, myös Canonin tulostusajoissa on edelleen huomattavia poikkeamia lineaariseen malliin verrattuna.





**Kuva 26.** Canon CLC 500+Fiery 125i:n tulostusaikakuvaaja, kun sekä EPS-sävykuvia, että runsaasti sivuja sisältävien tiedostojen tulostusajat on poistettu aineistosta.

## 6.2 Neuroverkkomallinnus käytännössä

### 6.2.1 Syöte- ja vasteparametrien käsittely

Kuten kirjallisuusosassa kappaleessa 4.2.4 todettiin, suuri merkitys neurolaskentaa sovellettaessa on syötteiden ja vasteiden esitystavalla. Esikokeissa neuroverkkoja yritettiin opettaa sekä elementtien datamäärillä, että niiden pohjalta lasketuilla prosentiosuuksilla kokonaisdatamäärästä. Saaduista tuloksista oli selvästi havaittavissa, että käytettäessä syöteinä elementtien prosentiosuuksia mallin ennustustarkkuus oli huomattavasti parempi. Tästä syystä malleissa päädyttiin käyttämään syöteinä absoluuttisten datamäärien sijasta elementtien datamäärien osuuksia kunkin tiedoston kokonaisdatamäärästä. Nämä prosentiosuudet on opetusaineiston osalta taulukoitu liitteessä 2 ja mallien testauksessa käytetyn aineiston osalta liitteessä 3 taulukossa 2.

Merkittävänä voidaan pitää myös havaintoa, että fonttien lukumäärä sivulla ei vaikuttanut havaittavasti mitattuihin tulostusaikoihin. Tämä poikkeama Pertti Mäkitalon tuloksiin saattaa johtua tässä työssä käytettyjen tulostimien työmuistin suuremmasta koosta. Lisäksi tähän saattaa vaikuttaa myös tulostimien ominaisuudet, eli esimerkiksi niiden tapaan käsitellä fontteja voi olla erilainen. Joka tapauksessa tässä työssä luovuttiin fonttien lukumäärän käyttämisestä tulostusaikoja selittävänä tekijänä. Jos on syytä olettaa, että fonttien lukumäärällä jonkin mallinnettavan tulostimen tapauksessa on vaikutusta tulostusaikoihin, voidaan tämä huomioida neuroverkkomallissa helposti. Opetusvaiheessa ainoastaan lisätään fonttien lukumäärä neuroverkon syötteeksi.

Neuroverkkomallien vasteparametrejä valittaessa päädyttiin käyttämään ainoastaan opetusaineiston tulostamisen yhteydessä mitattuja tiedostokohtaisia tulostusaikoja. Katsottiin, että kapasiteettia on turha ottaa neuroverkkomallin vasteeksi, koska se on muutenkin helppo laskettavissa sivumäärän ja ennustetun tulostusajan perusteella. Lisäksi tämän tyyppisessä mallintamisessa kapasiteetti on vertailukelpoinen mitta ainoastaan siinä

tapauksessa, että vertailtavilla järjestelmillä tulostetaan täsmälleen sama aineisto. Näin ollen tässä diplomityössä kehitetyt neuroverkkomallit ennustavat ainoastaan tietyn tiedoston tulostusaikaa ja jos tulostimen tai tulostusjärjestelmän kapasiteetti halutaan selvittää, on se laskettava erillisenä työvaiheena.

## 6.2.2 Käytettävän menetelmän valitseminen

Mallinnuksessa käytettäväksi menetelmäksi valittiin "ohjattu oppiminen". Jos tarkastellaan kirjallisuusosassa kappaleessa 4.2.2 esiteltyjä menetelmävaihtoehtoja, on tehty valinta helposti perusteltavissa. Ensinnäkin kiinteäpainoisten verkkojen tapauksessa ei oppimista varsinaisesti tapahdu lainkaan, vaan syötearvoja verrataan ennalta määriteltäviin malleihin. Tämän tyyppisen menetelmän soveltaminen tässä työssä edellyttäisi siis kaikkien mahdollisten tulostettavien tiedostojen rakenteen tallentamista muistiin, mikä on mahdolliton tehtävä. Itseoppivien verkkojen tapauksessa taas ei opetusvaiheessa käytetä vasteita lainkaan. Verkkojen avulla pyritään ainoastaan etsimään erilaisia lainalaisuuksia ja sääntöjä syötemateriaalista. Koska tämän työn ensisijaisena tarkoituksena oli pystyä ennustamaan järjestelmän suorituskykyä, ei myöskään tämän tyyppisen mallinnusmenetelmän hyödyntäminen tule kysymykseen. Ohjattu oppiminen soveltuu tässä työssä suoritettavaan mallintamiseen täsmälleen. Tavoitteena on selvittää tiettyjen toisiinsa liittyvien syötteiden (materiaali- ja järjestelmämuuttujat) ja vasteiden (tulostusajat) väliset riippuvuudet.

## 6.2.3 Oppimisalgoritmin valitseminen

Neuroverkkomalleja muodostettaessa käytettiin Matlab 4.2c.1 matriisilaskuohjelmaa, johon oli hankittu Neural Networks Toolbox v2.0b. Kuten edellä todettiin, päädyttiin mallinnuksessa hyödyntämään ohjattuun oppimiseen perustuvaa verkkoa. Tunnetuin ja myös eniten hyödynnetty tämän tyyppisen verkon opettamiseen soveltuva algoritmi on niin kutsuttu Backpropagation-algoritmi. Neural Network Toolbox sisältää useita versioita tästä algoritmista. Näistä päädyttiin esikokeiden perusteella käyttämään algoritmia, joka hyödyntää niin sanottua *momenttia* ja *adaptiivista oppimisnopeutta*. Momentin käyttäminen vähentää "Backpropagaation" herkkyyttä havaita pieniä yksityiskohtia virhetasosta. Tästä on apua kun pyritään välttämään juuttumista paikallisiin "minimikuoppiin". Adaptiivista oppimisnopeutta puolestaan käytetään oppimisnopeus-parametrin arvon säätämiseen opetuskerta-kohtaisesti. Algoritmi pyrkii tasaiseen oppimisnopeuteen säätämällä oppimisnopeus-parametrin arvoa verkon kulloisenkin tilan mukaan, eli käytännössä se pyritään pitämään mahdollisimman suurena läpi koko opetuksen. Näin oppimista voidaan nopeuttaa huomattavasti.

Backpropagation-verkkojen opetusalgoritmi perustuu niin sanottuun Widrow-Hoff opetusalgoritmiin. Tässä menetelmässä pyritään minimoimaan kunkin neuronikerroksen neuronien tekemästä virheestä laskettu neliösummavirhe. Tämä osoittautui kuitenkin yllättävän kehnoksi mitaksi, kun opetettuja verkkoja ryhdyttiin testaamaan tulostetulla testiaineistolla. Neuroverkkojen, joiden neliösummavirhe oli saatu minimoitua 0.001:een, ennusteet saattoivat heittää satoja, jopa tuhansia prosentteja kun taas neuroverkoilla, joiden neliösummavirhe oli niinkin korkea kuin 0.1 päästiin monta kertaluokkaa tarkempiin ennusteisiin. Tästä syystä verkkojen ennustustarkkuutta päädyttiin testaamaan tulostetulla testimateriaalilla. Tämä tarkoittaa käytännössä, että jokaiselle valmiiksi opetetulle verkolle syötettiin testimateriaalia kuvaavat muuttujat ja laskettiin verkon ennusteiden virhe. Verkko jonka ennusteiden virheet olivat pienimmät, valittiin lopulliseksi järjestelmää kuvaavaksi neuroverkkomalliksi.



## 6.2.4 Neuroverkkoarkkitehtuurin optimointi

Siinä vaiheessa kun järjestelmää kuvaavat syöte-vaste-parit<sup>1</sup> on muodostettu ja käytettävä neurolaskentamenetelmä sekä oppimisalgoritmi on valittu, voidaan ryhtyä pohtimaan varsinaisen neuroverkkoarakenteen muodostamista. Ensinnäkin on päätettävä, kuinka monta neuronikerrosta muodostettava verkko käsittää ja kuinka monta neuronua kukin näistä neuronikerroksista sisältää. Näiden valintojen tekemiseen ei ole käytössä mitään luotettavaa menetelmää, ainoastaan joitakin hyvin yleisellä tasolla määriteltyjä sääntöjä, joista osa on esitelty kirjallisuusosassa kappaleessa 4.2.4. Nämä tarjoavat kuitenkin ainoastaan lähtökohdan rakenteen suunnittelulle eli ideaalinen verkkoarkkitehtuuri joudutaan yleensä etsimään yrityksen ja erehdyksen kautta. Verkkoa opetettaessa vaihdellaan neuronikerrosten ja neuronien lukumäärää opetuskerrasta toiseen. Näin muodostettuja verkkoarkkitehtureja testataan opetusaineistoon kuulumattomalla materiaalilla ja pyritään löytämään paras mahdollinen rakenne.

Yllä esitettyjen tekijöiden määrittäminen muodostaa kuitenkin vasta yhden osan arkkitehtuurien muodostamisesta. Lisäksi on valittava neuronikerroksissa käytettävien aktivaatiofunktioiden tyypit. Tähän valintaan vaikuttaa syöte- ja vasteparametreille valitut esitysmuodot. Lisäksi on otettava huomioon, mitä verkolla aiotaan kuvata. Koska riippuvuudet on jo todettu epälineaariseksi, pyrittiin tämä seikka huomioimaan neuroverkkoarkkitehtuuria valittaessa. Parhaiten tämän tyyppiseen mallintamiseen soveltuvaksi on matemaattisesti osoitettu arkkitehtuuri, joka muodostuu yhdestä "sigmoid"-piilokerroksesta ja lineaarisesta vastekerroksesta. Tämän tyyppisellä verkolla voidaan teoriassa mallintaa mikä tahansa mielivaltainen funktio. Edellytyksenä on, että piilokerroksen neuronien lukumäärä on optimaalinen. Lisäksi järjestelmää kuvaavien syötteiden tulee olla siten valitut, että niiden avulla on mahdollista johtaa vasteiden arvot. Syötteiden ja vasteiden esitysmuoto on myös sovitettava käytettävälle rakenteelle ja oppimisalgoritmillemme, eli malli ei välttämättä pysty oppimaan riippuvuuksia mielivaltaisessa muodossa olevasta materiaalista.

Kun piilokerroksien neuronien lukumäärää ryhdytään optimoimaan on otettava huomioon joitakin tärkeitä seikkoja. Neuronien määrä on helposti valittavissa siten, että opetuksen onnistumisen mittana käytetty neliösummavirhe saadaan nopeasti minimoitua. Joidenkin opetusalgoritmien avulla verkko on mahdollista opettaa jopa siten, että neliösummavirhe on nolla. Tämän tyyppinen verkko pystyy ennustamaan opetusaineiston syöte-vaste-parit täysin virheettömästi. Tällaiseen tulokseen pääseminen edellyttää ainoastaan piilokerroksen neuronien lukumäärän kasvattamista riittävän suureksi. Matemaattisessa mielessä verkolle sallitaan tällöin enemmän vapausasteita. Tällöin ei kuitenkaan välttämättä ole kysymys siitä, että verkko olisi täydellinen ja kaikki sen ennusteet olisivat virheettömiä. Itse asiassa yleensä tilanne on täysin päinvastainen.

Kun neuroverkkoarakenteen sisältämien piilokerrosten neuronien lukumäärä kasvatetaan riittävän suureksi, malli pystyy oppimaan käytetyn opetusaineiston ulkoa. Tällöin kuitenkin menetetään neuroverkkojen yksi tärkeimmistä ominaisuuksista, eli kyky yleistää opittuja sääntöjä ennenkokemattomien tilanteiden ennustamiseen. Tästä syystä tavoitteena kannattaa aina pitää, että neuronien lukumäärä verkossa pidetään mahdollisimman pienenä. Tärkeintä on löytää eräänlainen tasapainotilanne neuronien lukumäärän ja verkon neliösummavirheen välillä. Jälleen kerran ainoa keino tämän tasapainotilanteen löytämiseksi on kokeileminen. Neuronien lukumäärän minimoimisella on myös toinen hyötynäkökohta eli mitä vähemmän verkossa on neuroneita, sitä vähemmän laskentaa tarvitaan ja sitä nopeampaa opettaminen on. Neuronien määrä ei kuitenkaan saa olla myöskään liian pieni, koska silloin verkko ei pysty oppimaan kaikkia riippuvuuksia syötteiden ja vasteiden välillä. Ensimmäistä tapausta kutsutaan ylioppimiseksi ja jälkimmäistä alioppimiseksi.

1. Tässä työssä muodostettiin useampia yhdistelmiä, koska haluttiin etsiä ideaalisin vaihtoehto.

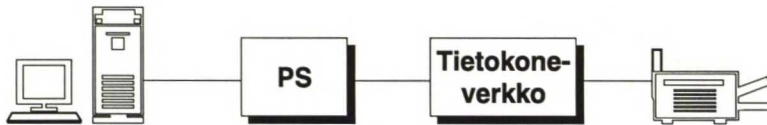


## 7

# YHDEN TULOSTIMEN SUORITUSKYKYMALLIT

## 7.1 Mallintamisen perusteet

Kuvassa 27 on esitetty verkkotulostusjärjestelmä, jolla kaikki luvussa 6 käsitelty opetusaineisto sekä tässä luvussa käytetty testimateriaali tulostettiin. Ympäristöä voidaan pitää tässä diplomityössä suoritettun mallintamisen perustana. Se muodostuu työasemasta, tietokoneverkosta ja tulostimesta. Mallintamisen tässä vaiheessa käytettiin ainoastaan yhtä työasemaa, jolta kaikki tulostettavat Postscript(PS)-tiedostot lähetettiin. Mallinnettavia tulostimia sen sijaan oli useampia (3), mutta ainoastaan yhdelle niistä tulostettiin kerrallaan. Tulostettava tiedosto siirrettiin tietokoneverkon kautta palvelimelle ja sieltä edelleen tulostimelle, jolla suoritettiin varsinainen tulostus. Tämä on siis perusympäristö, jossa mallinnetaan järjestelmään kuuluvien yksittäisten tulostimien suorituskyky.



**Kuva 27.** Perusmalli verkkotulostusjärjestelmästä, joka muodostuu työasemasta, tietokoneverkosta ja tulostimesta.

Kun jäljempänä tässä luvussa kuvattuja malleja ryhdyttiin muodostamaan kappaleessa 6.2.4 valitun neuroverkkorakenteen<sup>1</sup> avulla havaittiin, että opetuksen aikana neliösumma- virheet olivat todella suuria riippumatta piilokerroksien ja niiden sisältämien neuronien määrästä. Myös mallien ennusteiden virheet olivat suuria. Ratkaisuksi ongelmaan osoittautui vastekerroksen lineaarisen aktivaatiofunktion korvaaminen logaritmisella sigmoidiaktivaatiofunktioilla. Tämän seurauksena jouduttiin vasteina käytetyt tulostusajat skaalaamaan 0 ja 1 välille. Kyseinen toimenpide ei sinänsä näy mallien käyttäjälle eli se toteutettiin osana mallin rakennetta. Opetusvaiheessa vasteet muodostetaan yksinkertaisesti jakamalla mitatut tulostusajat 10.000:lla ja kun malleja käytetään tulostusaikojen ennustamiseen niiden ennustamat tulostusajat vastaavasti kerrotaan 10.000:lla. Tämä konsepti toimii ainoastaan siinä tapauksessa, että kaikki tulostusajat ovat alle 10.000 sekuntia ( $\approx 167$  min eli yli 2,5 h).

Kun neuroverkkoja ryhdytään opettamaan, suuri merkitys mallinnuksen onnistumiselle on myös sillä, missä järjestyksessä syöte-vaste-vektoriparit käsitellään. Ensimmäinen käsitelty opetusvektori esimerkiksi määrittää ikäänkuin aloituspisteen, josta mallia lähdetään tarkentamaan. Jos tämä piste on huonosti valittu, ei malli välttämättä opi mitään tai se

1. "Sigmoid"-piilokerros ja lineaarinen vastekerros.



ajautuu paikalliseen virheminiimiin, eikä ikinä löydä globaalia virheminiimiä. Matlabin neuroverkko-ohjelmassa tällaisia tilanteita pyritään välttämään siten, että opetuksen alussa ja myös sen aikana algoritmi valitsee syöte-vastevektoriparit sattumanvaraisessa järjestyksessä opetusaineistosta. Tästä on seurauksena, että vaikka mitään parametrejä itse algoritmilla ei muuteta, jokaisella opetuskerralla saadaan uusi, aiemmista poikkeava malli. Tästä syystä jokainen muodostettu arkkitehtuuri opetettiin kymmenen kertaa täsmälleen samoilla algoritmiparametreilla sen varmistamiseksi, että huono ennustustarkkuus ei johdu ainoastaan esimerkiksi huonosta aloituspisteestä. Näistä kymmenestä tiettyä arkkitehtuuria edustavasta mallista valittiin se, jonka ennusteen virhe oli kaikkein pienin. Näiden muodostettujen neuroverkkomallien virheitä verrattiin edelleen keskenään lopullisen, parhaan tuloksen antavan verkon löytämiseksi.

Mitattujen ja mallien avulla ennustettujen tulostusaikojen välisestä erosta laskettiin mallien tekemän virheen suuruus. Virheprosenttien laskemisessa käytettiin seuraavaa kaavaa:

$$\text{virhe-}\% [\text{tulostusaika}] = \frac{|T_e - T_m|}{T_m} \quad (3)$$

jossa  $T_e$  on ennustettu tulostusaika sekunneissa[s] ja  
 $T_m$  on mitattu tulostusaika sekunneissa [s].

Näistä tiedostokohtaisista virheistä laskettiin edelleen mallien tekemä keskimääräinen virhe. Tämä todettiin kuitenkin liian puutteelliseksi mitaksi mallien luotettavuutta arvioitaessa, joten lisäksi virheprosentista laskettiin niiden otoskeskihajonta ja otosvarianssi. Otoskeskihajonta kuvaa sitä, kuinka paljon yksittäisten ennusteiden virheet poikkeavat keskimääräisestä virheestä. Otosvarianssin avulla voidaan puolestaan analysoida laskettujen virheiden välistä keskinäistä hajontaa. Näitä kolmea tunnuslukua voidaan pitää ainoina järkevinä ennusteiden luotettavuuden mittoina, koska neuroverkkojen tapauksessa ei ole käytössä menetelmiä mallien tilastollisen luotettavuuden tutkimiseksi.

## 7.2 Testitulostimien suorituskykymallit

Tässä kappaleessa pyritään mallintamaan kukin testitulostuksissa käytetyistä tulostimista omalla neuroverkollaan. Opetusaineiston tulostusaikojen analysoinnin perusteella voidaan todeta, että vaikka EPS-sävykuvien osuus tiedoston datamäärästä ja lisäksi sivumäärä Canon CLC 500+Fiery 125i:n tapauksessa olivat selvästi merkittävimpiä lineaarisista malleista poikkeamia aiheuttavista tekijöistä, eivät ne välttämättä yksin riitä selittämään kaikkia vaihteluita. Todennäköistä on, että mitä tarkemmaksi tiedostojen sisältökuvaus muodostetaan, sitä tarkempi on myös opetettu malli. Tämän hypoteesin tutkimiseksi muodostettiin neljä mallia, joissa materiaalia kuvaavia opetusparametrejä lisättiin asteittain. Seuraavassa on käyty läpi näiden neljän perusmallin opetusparametrit.

- MALLI 1: Tiedoston koko [Mt].
- MALLI 2: Tiedoston koko [Mt], sivumäärä [kpl].
- MALLI 3: Tiedoston koko [Mt], sivumäärä [kpl], sävykuvien osuus datamäärästä [%], viivakuvien osuus datamäärästä [%] ja tekstin osuus datamäärästä [%].
- MALLI 4: Tiedoston koko [Mt], sivumäärä [kpl], BMP-, EPS-, ja TIFF-sävykuvien osuudet toisistaan erotettuna [%], CGM-, EPS- ja WMF-viivakuvien osuudet toisistaan erotettuna [%] ja tekstin osuus datamäärästä [%].

Malleja opetettaessa käytetty perusarkkitehtuuri muodostuu siis tangentiaalisesta "sigmoid"-piilokerroksesta, jonka neuronimäärää varioidaan ja logaritmisesta "sigmoid"-vaste-kerroksesta. Näin ollen syötteet skaalattiin -1...1 ja vasteet 0...1 välille. Lukuisten testien



jälkeen päädyttiin mallissa 1 käyttämään kahta, mallissa 3 kolmea ja mallissa 4 viittä piilokerroksen neuronია. Piilokerroksen neuronien tai itse piilokerrosten määrän lisääminen näiden kolmen ensimmäisen mallin tapauksessa johti ainoastaan ylioppimiseen. Tästä syystä tyydyttiin käyttämään pelkästään yhtä piilokerrosta ja neuronien määrä pidettiin näin alhaisena. Neuroverkko löytää tällä tavoin parhaan mahdollisen approksimaation ja jättää ikäänkuin eniten poikkeavat syöte-vasteparit huomiotta. Sen sijaan mallin 4 tapauksessa havaittiin selvästi sen tarkkuuden paranevan piilokerroksia ja niiden sisältämien neuronien määrää lisättäessä, eli näin ollen näitä toimenpiteitä voidaan pitää perusteltuina. Lopullisessa arkkitehtuurissa päädyttiin käyttämään kahta logaritmista "sigmoid"-piilokerrosta, joista kumpikin sisältää yksitoista neuronია. Vastekerroksen aktivaatiofunktio on edelleen logaritmisen "sigmoid"-funktio. Seuraavassa on muodostettuja malleja analysoitu HP:N ja IBM:n tulostimien osalta yhdessä, koska niiden tulokset ovat hyvin samansuuntaisia, vaikka poikkeavuuksiakin löytyy. Canon CLC 500+Fiery 125i:n tapauksessa tulokset olivat niin paljon edellisistä poikkeavia, että ne analysoitiin omana kokonaisuutenaan.

### HP LaserJet 4/4M+ PS ja IBM LaserPrinter 4029 PS17

Taulukossa 5 on esitetty HP LaserJet 4/4M+ PS:llä ja taulukossa 6 IBM LaserPrinter 4029 PS 17:llä mitatut testiaineiston tulostusajat, sekä aiemmin kuvatuilla neuroverkkomalleilla 1-4 ennustetut tulostusajat. Ennusteet on kunkin mallin osalta esitetty omassa sarakkeessaan. Samaan yhteyteen on liitetty todellisen ja ennustetun tulostusajan erotuksesta laskettu virheprosentti. Kunkin mallin syöteinä käytetyt muuttujat on kuvattu edellisessä kappaleessa. Periaatteena kuitenkin on, että siirryttäessä mallista 1 malliin 4, testiaineiston sisällön kuvaus tarkentuu koko ajan. Kunkin tiedoston kohdalla lasketuista virheprosentista on lisäksi laskettu keskimääräinen virhe, virheiden otoskeskihajonta ja otosvarianssi. Nämä on esitetty kunkin sarakkeen kohdalla taulukon alapuolella.

Kun saatuja tuloksia ryhdytään analysoimaan tarkemmin ensimmäinen havainto on, että keskimääräiset virheet ovat kummankin tulostimen tapauksessa käytännön sovelluksia ajatellen aivan liian suuria. Tarkempi tarkastelu kuitenkin osoittaa, että tilanne ei ole aivan niin toivoton kuin aluksi voitaisiin kuvitella. Huomiota kannattaa kiinnittää myös siihen, että vaikka IBM:n ja HP:n tulostimet ovat toteutusarkkitehtuuriltaan hyvin paljon toistensa kaltaisia, poikkeavat suorituskykyennusteet toisistaan. Keskimääräiset virheprosentit ovat IBM:n tapauksessa suurempia kuin HP:n tapauksessa. Suurimmat virhelähteet ovat kuitenkin täsmälleen samat, ja myös trendit ovat kummankin tulostimen tapauksessa hyvin samansuuntaisia joitain yksittäisiä tapauksia lukuunottamatta. Erojen voidaan olettaa johtuvan IBM:n tulostusaikojen suuremmasta hajonnasta eli ne vaihtelevat 104:n ja 8876:n sekunnin välillä, kun HP:n tapauksessa vastaava minimiarvo on 54 ja maksimiarvo puolestaan 3714 sekuntia. IBM:n kohdalla mallien tekemät virheet ovat suhteessa suurempia kuin HP:n.

Mallin 1 tapauksessa, jossa syöteenä käytettiin ainoastaan tulostettavan tiedoston kokoa, suurimmat virheprosentit aiheutuvat tiedostoista, jotka sisältävät ainoastaan EPS-sävykuvia (merkitty taulukoihin punaisella). Näiden tiedostojen ollessa kyseessä virheprosentit ovat todella suuria (>110%, IBM:n tapauksessa jopa 366%). Kuten edellisessä luvussa opeusaineistoa analysoitaessa havaittiin, juuri nämä tiedostot aiheuttivat myös lineaaristen mallien kohdalla suurimmat poikkeamat. Tätä olettamusta tukee myös seikka, että ennustetut tulostusajat näiden tiedostojen tapauksessa ovat nimenomaan suurempia kuin todellisuudessa mitatut tulostusajat. Merkillinen poikkeama tästä säännöstä on testiaineiston ensimmäinen tiedosto eli *wmfeps.ps*, joka myös sisältää ainoastaan EPS-sävykuvia, mutta jonka tulostusaikaennustetta voidaan pitää melko tarkkana. Tulostusvaiheessa todettiin empiirisesti, että kyseisen tiedoston sisältämien WMF-viivakuvien tulostaminen kesti merkittävästi kauemmin kuin vastaavien kuvien tulostaminen muissa testiaineiston tiedostoissa, joissa oli käytetty muita viivakuvaformaatteja.



Kysymys näyttäisi olevan siitä, että vaikka EPS-sävykuvat tulostuvat muita sävykuvia nopeammin, WMF-viivakuvien pidempi tulostusaika puolestaan kompensoi näin voitettua aikaa ja lopputuloksena syntyy näin pieni, mutta kuitenkin valheellinen virheprosentti. Huomiota kannattaa kiinnittää myös viimeiseen tiedostoon, jonka virheprosentti on niinkin suuri kuin 26% vaikka se sisältää ainoastaan tekstiä. Ainoa järkevä selitys tälle saattaisi löytyä sen kaksipalstaisuudesta, mutta ilmiö vaatii lisätutkimista ennen kuin johtopäätöksiä voidaan varmuudella tehdä. Muiden tiedostojen tapauksessa virheprosentit ovat siedettäviä eli alle 10% luokkaa mikä viittaa siihen, että neuroverkot todella soveltuvat tämän tyyppisten järjestelmien mallintamiseen.

**Taulukko 5.** HP LaserJet 4/4M+ PS:n todelliset ja neuroverkkomalleilla ennustetut tulostusajat. Virheprosentit on laskettu todellisten ja ennustettujen arvojen välisestä erosta kaavalla 3.

Tiedoston nimi	Tulostusaika [s]	Malli 1		Malli 2		Malli 3		Malli 4	
		Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]
wmfeps.ps	2884	3045	6 %	2946	2 %	2693	7 %	3184	10 %
epseps.ps	1387	2987	115 %	2901	109 %	2637	90 %	992	28 %
cgmeeps.ps	1429	2996	110 %	2910	104 %	2652	86 %	1944	36 %
epsbmp.ps	3309	2985	10 %	2899	12 %	2632	20 %	2751	17 %
epstiff.ps	3303	2985	10 %	2899	12 %	2630	20 %	2003	39 %
grakbmp.ps	1767	1785	1 %	1710	3 %	1550	12 %	2737	55 %
grakeps.ps	613	1850	202 %	1789	192 %	1652	169 %	729	19 %
graktiff.ps	1775	1785	1 %	1710	4 %	1534	14 %	2544	43 %
opetus_o.ps	1057	1337	26 %	657	38 %	1143	8 %	419	60 %
Keskimääräinen virhe		53 %		53 %		47 %		34 %	
Virheen otoskeskihajonta		72 %		67 %		56 %		17 %	
Virheen otosvarianssi		52 %		45 %		32 %		3 %	

Malli 2 antoi hyvin samansuuntaisia tuloksia kuin malli 1 mikä olikin odotettavissa, koska ainoa lisätty syöte oli tiedoston sivumäärä. Sivumäärällä ei edellisessä kappaleessa HP:n ja IBM:n tulostimien tapauksessa havaittu olevan merkittävää vaikutusta muodostuviin tulostusaikoihin, ja tämä näkyy myös mallin ennusteiden tarkkuudesta eli parannusta malliin 1 tapahtui tuskin lainkaan. Virheet ovat lähes samaa luokkaa. Mallin 3:n tapauksessa parannusta voidaan katsoa tapahtuneen jonkin verran, mutta edelleen suurimmat virheet aiheutuvat EPS-sävykuvia sisältävistä tiedostoista. Tämä on sinänsä loogista, koska jaottelu sävy- ja viivakuviin ei vielä kerro näiden kategorioiden sisäisestä formaattijakaumasta mitään ja juuri tällä on oletettavasti suurin vaikutus muodostuviin tulostusaikoihin. Ennusteiden tarkkuudet heikkenivät jonkin verran myös muiden tiedostojen kohdalla ja tämä viittaa siihen, että verkko ei enää pysty opetusmateriaalin perusteella löytämään riippuvuuksia syötteiden ja vasteiden välille samalla tarkkuudella kuin aiemmissa tapauksissa, jolloin syötteitä oli vähemmän. Näin ollen jo tässä vaiheessa voidaan todeta, että opetusmateriaalia on mallinnettavan järjestelmän monimutkaisuuteen nähden liian vähän.



Mallin 4 tapauksessa tapahtui radikaali muutos kolmeen aiempaan malliin verrattuna eli EPS-sävykuvia sisältävien tiedostojen virheprosentit putosivat selvästi. Samalla myös muiden tiedostojen virheprosentit nousivat eli verkko ikäänkuin hakee jonkinasteista keskiarvokuvausta aineistosta. Tämä johtuu hyvin todennäköisesti siitä, että opetusmateriaalia ei ole riittävästi kaikkein monimutkaisempien riippuvuuksien mallintamiseen. Myös *wmfeps.ps* tiedoston virheprosentti lähestyi muiden EPS-sävykuvia sisältävien tiedostojen virheprosentteja, vaikka se edelleen on niitä selvästi alhaisempi. Tämän yksittäisen tapausten pohjalta ilmiötä ei varmuudella voida selittää, mutta todennäköinen syy on, että neuroverkkomalli ei käytetyn määrällisesti rajallisen opetusaineiston perusteella kyennyt oppimaan WMF-viivakuvien vaikutusta muodostuviin tulostusaikoihin.

**Taulukko 6.** IBM LaserPrinter 4029 PS17:n todelliset ja neuroverkkomalleilla ennustetut tulostusajat. Virheprosentit on laskettu todellisten ja ennustettujen arvojen välisestä erosta kaavalla 3.

Tiedoston nimi	Tulostusaika [s]	Malli 1		Malli 2		Malli 3		Malli 4	
		Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]
wmfeps.ps	5470	6791	24 %	6348	16 %	8483	55 %	4040	26 %
epseps.ps	2536	6674	163 %	6275	147 %	4079	61 %	1447	43 %
cgmeeps.ps	2695	6687	148 %	6290	133 %	4163	54 %	1305	52 %
epsbmp.ps	7908	6671	16 %	6272	21 %	4061	49 %	3527	55 %
epstiff.ps	7937	6671	16 %	6272	21 %	4026	49 %	1492	81 %
grakbmp.ps	4171	4165	0 %	4207	1 %	3220	23 %	6286	51 %
grakeps.ps	932	4343	366 %	4415	374 %	3564	282 %	1866	100 %
graktiff.ps	4200	4165	1 %	4207	0 %	3310	21 %	2800	33 %
opetus_o.ps	2063	2963	44 %	2875	39 %	414	80 %	945	54 %
Keskimääräinen virhe		86 %		84 %		75 %		55 %	
Virheen otoskeskihajonta		122 %		122 %		80 %		23 %	
Virheen otosvarianssi		148 %		149 %		64 %		5 %	

Canon CLC500+Fiery 125i

Taulukossa 7 on esitetty Canon CLC 500+Fiery125i:llä tulostettujen testitiedostojen tulostusajat ja neljän mallin avulla ennustetut ajat. Käsittely on täysin vastaava kuin kahden edellä esitetyn tulostimen kohdalla eli kaikkien mallien osalta on laskettu tiedostokohtaisista virheprosentista kunkin mallin keskimääräinen virhe, virheiden otoskeskihajonta ja otosvarianssi. Tulosten analysointi katsottiin tarpeelliseksi erottaa, koska HP:n ja IBM:n tulostimille laadittujen mallien ennusteet poikkesivat niin selvästi Canonille muodostettujen mallien ennusteista. Selitys näin suurille poikkeamille löytyy luultavimmin tulostimien erilaisesta toteutusarkkitehtuurista eli siitä, että Canonin tapauksessa rasterointiprosessori on erotettu omaksi kokonaisuudekseen varsinaisesta tulostinyksiköstä. Lisäksi kun tuloksia analysoitiin luvussa 6 havaittiin, että sivumäärällä on aiemmista kahdesta tulostimesta poiketen merkittävä vaikutus muodostuviin tulostusaikoihin.

Tarkasteltaessa mallin 1 ennusteita havaitaan, että virheprosentit ovat jopa yllättävän pieniä. Erityisesti kahden aiemman tulostimen tapauksessa eniten virhettä aiheuttaneiden



EPS-sävykuvia sisältävien tiedostojen ennusteet ovat tässä tapauksessa jopa tarkempia kuin muiden tiedostojen. Ainoa järkevä selitys tämän tyyppiselle vääristymälle löytyy EPS-sävykuvien ja sivumäärän yhteisvaikutuksesta. EPS-sävykuvien käsittely on tehokkaampaa kuin tiedoston koko antaa olettaa, mutta koska tiedostot sisältävät runsaasti sivuja on sivukohtainen tulostusaika puolestaan pidempi kuin mihin tiedoston koko viittaa. Lopputuloksena saadaan taulukosta nähtävät suhteellisen pienet, mutta vääristyneet virheprosentit. Tätä väittämää tukee myös seikka, että mallien 2 ja 3 tapauksessa virheprosentit kasvavat päinvastoin kuin kävi aiempien tulostinmallien tapauksessa. Tämä johtuu siitä, että kun mallien syötteitä lisätään ne ikäänkuin havaitsevat todelliset vaikuttavat tekijät ja pyrkivät korjaamaan ennusteita niiden mukaan, johon taas ei käytetyn opetusmateriaalin välittämä informaatio riitä. Mallissa 4 tämä puute hieman korjaantuu, koska syötteet sisältävät riittävästi selittäviä tekijöitä tarkempien johtopäätösten tekemiseksi. Edellä esitetystä voidaan päätellä, että vaikka tulokset poikkeavat edellisten tulostinmallien tuloksista, tukevat ne kokonaisuutena tulkittuina edellisten tulostimien tapauksessa tehtyjä johtopäätöksiä.

**Taulukko 7.** Canon CLC 500+Fiern 125i:n todelliset ja neuroverkkomalleilla ennustetut tulostusajat. Virheprosentit on laskettu todellisten ja ennustettujen arvojen välisestä eroista kaavalla 3.

		Malli 1		Malli 2		Malli 3		Malli 4	
Tiedoston nimi	Tulostusaika [s]	Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]	Ennuste [s]	Virhe [%]
wmfeps.ps	5927	6303	6 %	6643	12 %	7680	30 %	4240	28 %
epseps.ps	4722	3876	18 %	6370	35 %	7534	60 %	4118	13 %
cgmeps.ps	4769	4006	16 %	6393	34 %	7548	58 %	3827	20 %
epsbmp.ps	5945	3851	35 %	6365	7 %	7533	27 %	4230	29 %
epstiff.ps	5953	3851	35 %	6365	7 %	7541	27 %	4038	32 %
grakbmp.ps	2109	1679	20 %	1869	11 %	1837	13 %	2421	15 %
grakeps.ps	1355	1721	27 %	1903	40 %	1860	37 %	1720	27 %
graktiff.ps	2114	1679	21 %	1869	12 %	1831	13 %	2282	8 %
opetus_o.ps	2476	1410	43 %	4919	99 %	4036	63 %	2937	19 %
Keskimääräinen virhe		25 %		29 %		36 %		21 %	
Virheen otoskeskihajonta		12 %		29 %		19 %		8 %	
Virheen otosvarianssi		1 %		9 %		4 %		1 %	

Canon CLC 500+Fiern 125i:n tapauksessa voidaan selvästi havaita opetusaineiston puutteellisuus, eli kun merkittäviä epälineaarisuutta aiheuttavia tekijöitä on useampia kuin yksi, opetettu malli ei pysty lähimainkaan tarkkoihin ennustuksiin. Ainoa mahdollisuus tarkentaa malleja on opetusaineistona käytetyn materiaalin lisääminen. Luultavimmin on myös tarpeen muodostaa lisää rakenteeltaan ja sisällöltään vaihtuvia tiedostoja riippuvuuksien selkeyttämiseksi.



## 7.3 Käsitellyt erikoistapaukset

Koska EPS-sävykuvien ja sivumäärän vaikutus tulostusaikoihin testitulostimien tapauksessa oli selvä, muodostettiin erillinen neuroverkkomalli sen tutkimiseksi kuinka tarkasti pelkästään näiden ominaisuuksien perusteella pystytään tietyllä tulostimella muodostuvaa tulostusaikaa ennustamaan. Mallin syötteinä käytettiin siis ainoastaan seuraavia muuttujia:

- tiedoston koko [Mt],
- sivumäärä [kpl] ja
- EPS-sävykuvien osuus kokonaisdatamäärästä [%].

Muodostettu neuroverkko on rakenteeltaan periaatteessa samanlainen kuin mallit 1-3 edellisessä kappaleessa. Se muodostuu syötekerroksesta, yhdestä piilokerroksesta ja vastekerroksesta. Myös aktivaatiofunktioiden osalta rakenne vastaa aiempia malleja eli piilokerroksessa käytetään tangentiaalista ja vastekerroksessa puolestaan logaritmista "sigmoid"-funktiota. Se poikkeaa niistä ainoastaan piilokerroksen neuronien lukumäärän osalta; tässä mallissa niitä on seitsemän. Huomattavaa on, että neuroverkkomallit opetettiin edelleen kunkin tulostimen kohdalla omana erillisenä kokonaisuutenaan eli kyseiset mallit eivät ole yleispäteviä. Malleja voidaan soveltaa siis ainoastaan sen tulostimen tulostusaikojen ennustamiseen, jolla neuroverkon opetukseen käytetty testiaineisto on tulostettu.

Kun mallia ryhdyttiin opettamaan tehtiin sama havainto kuin aiempien mallien kohdalla eli piilokerroksien lisääminen tai niiden sisältämien neuronien määrän kasvattaminen yli seitsemän johti ainoastaan ylioppimiseen. Taulukkoon 8 on koottu kyseisten tulostinkohdainten mallien avulla ennustetut tulostusajat kunkin testitulostimen osalta. Vertailun helpottamiseksi on ennusteiden rinnalla esitetty myös varsinaiset mitatut tulostusajat. Todellisten ja ennustettujen tulostusaikojen perusteella on laskettu mallien tekemät virheet täsmälleen samalla tavoin kuin edellä käyttäen hyväksi kaavaa 3. Näistä tiedostokohtaisista virheistä on edelleen laskettu mallien tekemä keskimääräinen virhe, sekä virheiden otoskeskihajonta ja otosvarianssi. Nämä tunnusluvut on esitetty kunkin mallin osalta taulukon alapuolella. Nämä arvot ovat täysin vertailukelpoisia edellisessä kappaleessa laskettujen arvojen kanssa ja niiden välillä voidaan suorittaa vertailua sen toteamiseksi, onko muodostetun mallin ennustustarkkuus parempi kuin aiempien mallien.

Laskettuja tunnuslukuja tarkasteltaessa voidaan selvästi havaita, että keskimääräiset virheet ovat suunnilleen samaa luokkaa kuin edellisessä kappaleessa esitettyjen yhdeksän syötteen mallien tapauksessa (MALLI 4). Sen sijaan virheiden otoskeskihajonnat ovat noin 1,5–3 kertaisia edellisessä kappaleessa opetettujen mallien tekemistä virheistä laskettuihin verrattuna. Myös otosvarianssit ovat 2,5–7 kertaisia. Näin ollen vaikka keskimääräiset virheet ovat hyvin lähellä edellisen kappaleen MALLIN 4 vastaavia arvoja, ei tässä kappaleessa muodostettuja malleja kuitenkaan voida pitää yhtä tarkkoina. Selvinä poikkeamina edellisen kappaleen MALLIN 4 ennusteista voidaan pitää taulukossa punaisella merkittyjä EPS-sävykuvia sisältävien tiedostojen tulostusaikaennusteita. Huolimatta siitä, että syötteinä käytettiin EPS-sävykuvatiedostojen suhteellista osuutta kokonaisdatamäärästä, muodostetut HP:n ja IBM:n mallit eivät selvästikään pystyneet ottamaan huomioon kyseisten elementtien osuuden vaikutuksia riittävällä tarkkuudella. Tähän saattaa olla syynä myös opetusaineiston vähyys.

Canonin tapauksessa ennusteet ovat kuitenkin hyvin samantyyppisiä kuin edellisessä kappaleessa taulukossa 7 kuvattujen MALLIEN 1-4 tulokset. Vaikka virheprosentit ovat suhteellisen pieniä, ei niitä voida pitää luotettavina, koska tulostusaikoihin merkittävästi vaikuttavia tekijöitä on useampia. Canonin tapauksessa nämä tekijät ovat EPS-sävykuvien osuus ja sivumäärä. Koska ne lisäksi ovat virhelähteitä, joiden vaikutukset ikäänkuin kompensoivat toisiaan, voidaan ennusteita pitää tietyllä tavalla vääristyneinä. Tämäkin johtuu



oletettavasti siitä, että käytetty opetusaineisto ei ollut riittävän laaja ja monipuolinen. Opetusaineistoa lisäämällä on luultavasti mahdollista opettaa myös tämän tyyppiset vähemmän syöteparametrejä sisältävät mallit huomattavasti tarkemmiksi.

**Taulukko 8.** Alla olevassa taulukossa on esitetty testiaineiston tulostusajat kunkin tulostimen osalta ja tulostinkohtaisten mallien avulla ennustetut tulostusajat. Malleissa käytettiin syötteinä ainoastaan tiedoston kokoa, sivumäärää ja EPS-sävykuvien osuutta kokonaisdatamäärästä. Virheprosentit on laskettu kaavalla 3.

	HP	Malli		IBM	Malli		Canon	Malli	
Tiedoston nimi	Tulostusaika [s]	Ennuste [s]	Virhe [%]	Tulostusaika [s]	Ennuste [s]	Virhe [%]	Tulostusaika [s]	Ennuste [s]	Virhe [%]
wmfeps.ps	2884	2551	12 %	5470	6577	20 %	5927	2538	57 %
epseps.ps	1387	2415	74 %	2536	6529	157 %	4722	2528	46 %
cgmeeps.ps	1429	2466	73 %	2695	6544	143 %	4769	2530	47 %
epsbmp.ps	3309	2504	24 %	7908	6568	17 %	5945	2535	57 %
epstiff.ps	3303	2504	24 %	7937	6568	17 %	5953	2535	57 %
grakbmp.ps	1767	1630	8 %	4171	5159	24 %	2109	2073	2 %
grakeps.ps	613	861	40 %	932	1485	59 %	1355	1344	1 %
graktiff.ps	1775	1630	8 %	4200	5159	23 %	2114	2073	2 %
opetus_o.ps	1057	490	54 %	2063	3367	63 %	2476	2164	13 %
Keskimääräinen virhe		35 %		58 %		31 %			
Virheen otoskeskihajonta		26 %		55 %		26 %			
Virheen otosvarianssi		7 %		30 %		7 %			

Merkittävää on, että vaikka tämän tyyppinen malli ei tarkkuudessa pärjääkään yhdeksän syötteen malleille (MALLI 4), antaa se huomattavasti luotettavampia ennusteita kuin edellisessä kappaleessa muodostetut MALLIT 1, 2 ja 3. Tämä pätee kaikkien kolmen testitulostimen tapauksessa. Tämän tyyppisiä malleja voitaisiin mahdollisesti soveltaa tapauksissa, joissa tulostettavaa materiaalia kuvaavaa informaatiota on käytössä rajoitetusti. Onhan selvää, että yhden ainoan sisältömuuttujan selvittäminen tiedostosta on helpompaa kuin MALLISSA 4 käytettyjen seitsemän sisältömuuttujan. On hyvin todennäköistä, että mallien opettamista jatkettaessa, päästään niillä ennustustarkkuuksiin, jotka ovat riittäviä myös hajautettujen järjestelmien malleja silmällä pitäen.

### 7.4 Yleinen yhden tulostimen malli

Koska suorituskykymalleja on tarkoitus hyödyntää myös tulostusjärjestelmiä suunniteltaessa ja mitoitettaessa muodostettiin yleinen tulostinmalli, jonka avulla on tarkoitus pystyä ennustamaan minkään tahansa järjestelmään liitettäväksi aiotun tulostimen suorituskykyä. Jos tämän tyyppinen malli todetaan toimivaksi, on tarpeetonta mallintaa jokaista järjestelmään liitettyä tulostinta erikseen, vaan riittää kun malli opetetaan riittävällä

määrällä tulostimia, joiden ominaisuudet poikkeavat toisistaan. Tässä tapauksessa yleisen mallin opetusparametreiksi valittiin edellä kuvatuista malleista tarkimman tuloksen antaneen syötteet. Koska paras tarkkuus saavutettiin malleilla jotka sisälsivät yhdeksän syötettä (MALLI 4), muodostettiin yleinen malli näiden syötteiden pohjalta. Näiden syötteiden lisäksi opetuksessa käytettiin tulostimien suorituskykyä kuvaavia suureita, joiden valintakriteeri oli, että niiden tulee olla helposti tulostimien teknisistä tiedoista selvitettävissä. Siihen, miten hyvin nämä parametrit mahdollisesti korreloivat tulostusaikojen kanssa ei tässä yhteydessä oteta kantaa. Edellä esitetyn kriteerin täyttivät seuraavat tulostinominaisuudet:

- koneistonopeus sivuina minuutissa [s/min],
- työmuistin koko megatavuina [Mt],
- resoluutio pisteinä tuumalla [dpi] ja
- prosessorin teho megahertseinä [MHz].

Testitulostimien osalta kyseiset muuttujat on esitetty kappaleessa 5.2 taulukossa 4. Kuten taulukosta havaitaan, tulostimet soveltuvat erittäin hyvin tämän tyyppisen yleisen mallin opettamiseen ja testaukseen, koska ne poikkeavat toisistaan kaikkien edellä mainittujen ominaisuuksien osalta. Tällä on positiivinen vaikutus nimenomaan muodostetun neuroverkon yleistysominaisuuksiin, eli mitä useampia poikkeavia muuttujia syötteiksi valitaan, sitä paremmin muodostettu malli pystyy reagoimaan järjestelmässä vaikuttaviin erityyppisiin muuttujiin. Aiemmissa malleissa on kuitenkin törmätty ongelmiin siinä vaiheessa, kun syötteiden määrä kasvaa liian suureksi ja tässä mallissahan niitä on kolmetoista (9+4). Täten on hyvin mahdollista, että käytössä olevan opetusmateriaalin avulla ei enää ole mahdollista selvittää näin monen aineisto- ja järjestelmämuuttujan välisiä riippuvuuksia. Mallin avulla ennustetut ja mitatut tulostusajat on esitetty taulukossa 9.

**Taulukko 9.** Yleisen tulostinmallin avulla ennustetut tulostusajat ja todellisiin tulostusaikoihin vertaamalla lasketut virheprosentit. Virheprosenttien laskemiseen käytettiin kaavaa 3.

	HP	Yleinen malli		IBM	Yleinen malli		Canon	Yleinen malli	
Tiedoston nimi	Tulostusaika [s]	Ennuste [s]	Virhe [%]	Tulostusaika [s]	Ennuste [s]	Virhe [%]	Tulostusaika [s]	Ennuste [s]	Virhe [%]
wmfeps.ps	2884	4242	47 %	5470	7120	30 %	5927	4582	23 %
epseps.ps	1387	2575	86 %	2536	4745	87 %	4722	3739	21 %
cgmepps.ps	1429	2774	94 %	2695	4967	84 %	4769	3689	23 %
epsbmp.ps	3309	2471	25 %	7908	4862	39 %	5945	3328	44 %
epstiff.ps	3303	2052	38 %	7937	4247	46 %	5953	3142	47 %
grakbmp.ps	1767	2298	30 %	4171	5871	41 %	2109	2391	13 %
grakeps.ps	613	1066	74 %	932	2172	133 %	1355	1426	5 %
graktiff.ps	1775	1490	16 %	4200	3691	12 %	2114	2114	0 %
opetus_o.ps	1057	807	24 %	2063	1453	30 %	2476	2128	14 %
Keskimääräinen virhe		48 %		56 %		21 %			
Virheen otoskeskihajonta		29 %		38 %		16 %			
Virheen otosvarianssi		8 %		15 %		3 %			



Testien kautta muodostettu yleiseksi tulostinmalliksi soveltuva neuroverkko rakentuu syötekerroksesta, kahdesta piilokerroksesta ja vastekerroksesta. Kumpikin piilokerroksesta sisältää 37 neuronia. Aktivaatiofunktioit ovat samat kuin edellisessä kappaleessa yhdeksän syötteen mallien tapauksessa eli kaksi piilokerrosta hyödyntävät tangentiaalista ja vastekerros logaritmistä "sigmoid"-funktioita. Tämän mallin kohdalla on huomattava, että koko opetus suoritettiin kaikkien tulostimien osalta yhdellä kertaa eli kaikki edellä taulukossa esitetyt ennusteet on laskettu samalla neuroverkolla. Opetusaineisto rakennettiin siten, että MALLISSA 4 käytetyt yhdeksän muuttujan syötevektorit kopioitiin kolmena kappaleena samaan tiedostoon. Kunkin kolmen vektorijoukon perään liitettiin edellä kuvatut tulostinkohtaiset muuttujat. Näin saatiin muodostettua  $13 \times 150$  syötematriisi eli 3 kertaa 50 kappaletta 13 muuttujan vektoreita. Vasteiden kohdalla toimittiin samalla tavoin eli kunkin tulostimen tulostusajat koottiin yhteen tiedostoon ja näin saatiin muodostettua  $1 \times 150$  kokoinen vastevektori.

Tarkasteltaessa keskimääräisiä virheprosentteja ja verrattaessa niitä aiempien mallien kohdalla laskettuihin voidaan havaita, että lukuunottamatta HP:n tulostinta, ne ovat jopa yllättävällä tarkkuudella samaa luokkaa kuin tulostinkohtaisilla malleilla. Otoskeskihajonnat ja otosvarianssit ovat hieman suurempia, mutta näitäkään eroja ei voida pitää merkitävän kokoisina kun otetaan huomioon että syötteiden määrä on mallissa jo kolmetoista. HP:n tapauksessa keskimääräinen virhe on hieman yli 10% suurempi kuin kappaleessa 7.2 MALLIN 4 ja edellisessä kappaleessa käsitellyn kolmen syötteen mallin tapauksessa. Tähän on vaikea löytää mitään järkevää selitystä käytössä olevan materiaalin perusteella. Eräs mahdollisuus on, että jos neuroverkkorakennetta olisi edelleen kehitetty ja opetuskerroja lisätty, olisi mahdollisesti löydetty malli, jossa kyseinen virhe olisi korjaantunut. Tässä työssä oli kuitenkin tarpeen rajoittaa mallien optimointiin käytettyä aikaa ja yllä esitetyt tulokset ovat parhaalla mahdollisella, näissä puitteissa rakennetulla mallilla ennustettuja.

Tuloksien perusteella voidaan kuitenkin olettaa, että neurolaskennan avulla on mahdollista rakentaa malli, joka ei vaadi tulostinkohtaista opettamista vaan tietyn rajatun ominaisuusiltaan toisistaan poikkeavan tulostinjoukon avulla voidaan kehittää malli, jonka avulla voidaan ennustaa minkä tahansa järjestelmään liitettävän tulostimen suorituskykyä jo ennen sen varsinaista fyysistä kytkemistä. Edellytyksenä on ainoastaan, että tulostimesta tunnetaan edellä kuvatut neljä ominaisuutta. Tämän tyyppinen yleinen tulostinmalli on erittäin hyödyllinen jo siinäkin mielessä, että järjestelmään kuuluvia tulostimia mallinnettaessa ei ole välttämätöntä tulostaa opetusmateriaalia kaikilla tulostimilla vaan ainoastaan riittävän monella, jotta tarvittavat riippuvuudet saadaan kuvattua. Näin säästetään sekä aikaa, että rahaa.

8

# HAJAUTETTUIJEN PAINATUSJÄRJESTELMIEN SUORITUSKYKYMALLIT

## 8.1 Hajautettujen järjestelmien mallintaminen

### 8.1.1 Mallintamisen lähtökohta

Alunperin tarkoituksena oli ennustaa myös useamman tulostimen muodostamien hajautettujen tulostusjärjestelmien suorituskykyä neuroverkkojen avulla. Kun otetaan huomioon aika, joka kuluu yhden arkkitehtuurin opetusmateriaalin tulostamiseen ja neuroverkon opettamiseen, ei tämä lähestymistapa tunnu kuitenkaan kovin järkevältä. Jos järjestelmä muodostuu esimerkiksi kolmesta tulostimesta, mutta tulostettaessa halutaan hyödyntää ainoastaan kahta, on tämä arkkitehtuuri mallinnettava erikseen. Lisäksi jos tulostimet poikkeavat ominaisuuksiltaan toisistaan, voidaan niistä edelleen muodostaa useampia yhdistelmiä. Kaikissa tapauksissa opetusmateriaali olisi tulostettava kokonaisuudessaan jokaisella tulostimella. Huomattavasti helpompaa on mallintaa hajautetun järjestelmän muodostavat tulostimet erikseen yksi kerrallaan kuten edellä tehtiin, ja pyrkiä ennustamaan näiden tulostusaikojen perusteella halutulla tavalla muodostetun arkkitehtuurin suorituskykyä. Tällöin opetusmateriaali tulostetaan jokaisella tulostimella ainoastaan kerran. Alla on esitetty tulostettavat sivumäärät kummassakin ratkaisumallissa. Esimerkkitapauksessa järjestelmän on oletettu muodostuvan kolmesta tulostimesta ja testimateriaalin on oletettu käsittävän 500 sivua.

Arkkitehtuurit erikseen	Sivumäärä	Tulostimet erikseen	Sivumäärä
Tulostin 1	500	Tulostin 1	500
Tulostimet 1 ja 2	1000	Tulostin 2	500
Tulostimet 1, 2 ja 3	1500	Tulostin 3	500
Tulostimet 2 ja 3	1000	–	–
Tulostimet 1 ja 3	1000	–	–
Yhteensä	5000	Yhteensä	1500

Kuten taulukoista havaitaan on tulostettava sivumäärä yli kolminkertainen siinä tapauksessa, että arkkitehtuurit mallinnettaisiin erikseen. Lisäksi tämä useamman mallinnettavan arkkitehtuurin lähestymistapa ei tarjoa mahdollisuutta hyödyntää yleisiä tulostinmalleja, joiden kehittäminen edellisessä luvussa tehtyjen kokeiden perusteella vaikuttaa mahdolliselta ja myös erittäin houkuttelevalta. Tässä luvussa käydään läpi erilaisia



järjestelmäarkkitehtuureja, joista jokaiselle on muodostettu oma malli tulostusaikojen ennustamista varten. Kovin laajoja järjestelmiä ei Teknillisen korkeakoulun Graafisen tekniikan laboratoriossa pystytty rakentamaan, johtuen tulostimien rajallisesta määrästä. Malleja muodostettaessa on kuitenkin pyritty pohtimaan myös niiden soveltuvuutta laajempien järjestelmien mallintamiseen.

Neuroverkkomallien opetusmateriaalia tulostettaessa havaittiin, että siirto työasemalta tulostusjonoon ja sieltä edelleen verkon kautta tulostimelle tapahtuu riittävällä tarkkuudella vakiollisella nopeudella. Siirtoon kulunut aika on suoraan riippuvainen siirrettävästä datamäärästä ja työaseman verkkokortin suorituskyvystä. Tällä perusteella voidaan olettaa, että jos pystytään määrittämään siirtoon kuluva aika ja ennustamaan neuroverkkomallien avulla yksittäisen tulostimen suorituskyky, pystytään muodostamaan matemaattisia malleja useamman tulostimen järjestelmän kokonaissuorituskyvyn määrittämiseksi. Luvussa 7 muodostettujen neuroverkkomallien ennustustarkkuus käytetyn opetusmateriaalin pohjalta on keskimäärin heikko, joten järjestelmän kokonaissuorituskykyä ennustavia malleja testattaessa käytettiin syötteinä mitattuja, ei ennustettuja tulostusajkoja. Tällä tavoin voidaan luotettavammin todeta, mikä on hajautettujen järjestelmien suorituskyvyn ennustamiseen kehitettyjen mallien tarkkuus ilman että neuroverkkomallien epätarkkuudet vääristävät tuloksia.

### 8.1.2 Tulostustapahtuman analysointi

Luvussa 7 muodostettujen yhden tulostimen mallien kannalta merkittävimpinä prosessivaiheina voidaan pitää seuraavia:

- siirto työasemalta tiedostopalvelimelle,
- sijoittaminen tulostusjonoon verkkolevyille,
- siirto tulostusjonosta tulostimen työmuistiin rasterointia varten,
- rasterointi ja
- varsinainen tulostus.

Tarkasteltaessa näitä prosessin vaiheita lähemmin voidaan päätellä, että suorituskyvyn kannalta ainoastaan sijoittaminen tulostusjonoon on täysin riippumaton siitä, mitä työasemaa tai tulostinta käytetään. Siirtoaikaan työasemalta palvelimelle vaikuttaa työaseman ja palvelimen verkkokorttien suorituskyky. Kun tiedosto siirretään palvelimelta tulostimelle, vaikuttaa siirtoon kuluvaan aikaan puolestaan palvelimen ja tulostimen verkkokorttien, sekä niiden välisen yhteyden suorituskyky. Lisäksi tässä vaiheessa suoritettavien siirtojen määrään vaikuttaa myös tulostimen työmuistin koko. Rasterointinopeus riippuu rasterointiprosessorin tehokkuudesta ja tulostusnopeus tulostimen koneistonopeudesta.

Koska neuroverkkomalleilla ennustettiin koko järjestelmän suorituskykyä, ei yllä kuvattuja prosessin osavaiheita testitulostusten yhteydessä mitenkään erotettu toisistaan. Koska lopullisena tavoitteena on ennustaa yksittäisten tulostimien suorituskykyennusteiden avulla useammasta työasemasta ja tulostimesta muodostuvien järjestelmien kokonaissuorituskykyä on pohdittava, mitkä prosessivaiheet tulisi laajempaa mallia ajatellen pystyä erottamaan yhden tulostimen suorituskykyennusteista. Jos tulostusympäristöä ajatellaan laajemmassa mittakaavassa, voidaan kaikki merkitykselliset järjestelmämuuttujat kiinnittää johonkin järjestelmän osakomponenteista. Itse asiassa tulostusjärjestelmä voidaan yksinkertaisesti jakaa kahteen osaan ja jakajana voidaan käyttää tiedosto/tulostinpalvelinta. Aika joka kuluu tulostettavan tiedoston lähettämiseen palvelimelle kuvaa työaseman suorituskykyä. Ennustetusta kokonaistulostusajasta jäljelle jäävä osuus on aina tulostin-kohtainen ja se kuvaa tulostimen suorituskykyä. Ainoa neuroverkkomallien ennustamasta kokonaistulostusajasta selvitettävä tekijä on siis aika joka kuluu tiedoston siirtämiseen työasemalta palvelimelle.



Varsinaisia hajautettujen painatusjärjestelmien malleja testattaessa käytettiin liitteessä 4 "tulostusaika"-sarakkeessa esitettyjä aikoja. Koska ainoa aiemmissa testitulostuksissa mitattu suure on "yhteensä"-sarakkeessa esitetty kokonaistulostusaika, tulee siitä pystyä erottamaan tiedoston siirtoon kulunut aika. Siirtoaika on laskettu tiedoston koon ja käytetyn työaseman siirtonopeuden perusteella, tähän palataan tarkemmin kappaleessa 8.1.3. "Tulostusaika"-sarakkeen arvot on siis laskettu vähentämällä "yhteensä"-sarakkeen arvosta laskettu siirtoaika. Kun syöteinä käytetään tulostusaikoja joista siirtoaika on vähennetty, voidaan malleissa huomioda tiedostojen lähettämiseen käytettyjen työasemien ominaisuudet riippumatta siitä, mitä työasemaa neuroverkkomallien opetusvaiheessa on käytetty. Tästä lähin puhuttaessa tiedostojen tulostusajoista tarkoitetaan siis nimenomaan aikoja, joista siirtoaika on vähennetty.

### 8.1.3 Käytettyjen työasemien ominaisuuksien määrittäminen

Kun tässä luvussa käytettyä materiaalia ryhdyttiin eri arkkitehtuureilla tulostamaan, mitattiin aiemmin suoritetuista tulostuksista poiketen tiedoston siirtämiseen kulunut aika. Tämä on välttämätöntä siitä syystä, että pystytään määrittämään käytettyjen työasemien suorituskyky. Työasemien osalta mitatut siirtoajat on esitetty varsinaisten tulostusaikojen yhteydessä liitteen 5 taulukoissa. Taulukoissa 1-4 on esitetty yhden työaseman arkkitehtuurien tulostusajat. Näissä arkkitehtuureissa on käytetty samaa työasemaa kuin edellisessä luvussa neuroverkkojen opetus- ja testimateriaalia tulostettaessa. Kyseisen työaseman keskimääräiseksi siirtonopeudeksi määritettiin mitattujen aikojen perusteella 20,88 kt/s. Tätä siirtonopeutta käytettiin myös laskettaessa liitteen 4 siirtoaikoja, jotka vähennettiin käytettyjen testitulostimien mitatuista kokonaistulostusajoista. Tähän käytettiin yksinkertaista kaavaa

$$\text{Siirtonopeus [s]} = \frac{\text{Tiedoston koko [kt]}}{\text{Siirtoon kulunut aika [s]}} \quad (4)$$

Taulukossa 5 puolestaan on esitetty 2-3 työaseman arkkitehtuurien tapauksessa mitatut tulostusajat. Kaiken kaikkiaan käytettiin siis neljää työasemaa, joihin vastedes viitataan ainoastaan juoksevilla numerolla. Edellä kuvattu työasema on siis numero 1 ja alla kuvattuihin tulostimiin viitataan numeroilla 2, 3 ja 4. Kunkin kolmen tulostimen osalta mitattiin siirtoajat vastaavalla tavalla kuin edellä testitulostusten yhteydessä. Näiden aikojen ja tiedostokokojen perusteella laskettiin kullekin työasemalle niiden suorituskykyä kuvaava keskimääräinen siirtonopeus. Nämä siirtonopeudet olivat:

- Työasema 2    16.44 kt/s
- Työasema 3    16.44 kt/s
- Työasema 4    13.97 kt/s

Itse asiassa tällä tavoin mitattu siirtoaika ei kuvaa suoranaisesti verkon suorituskykyä, vaan kysymyksessä on nopeus, jolla tiedostopalvelin pystyy sijoittamaan työaseman lähettämiä datapaketteja tulostusjonoon eli käytännössä verkkolevylle. Näin ollen verkon siirtonopeuden tunteminen ei kerro koko totuutta muodostuvista siirtoajoista, vaan niihin vaikuttavat verkkokorttien ja verkkojen ominaisuuksien lisäksi palvelimen, sen sisäisten väylien, verkkolevyn ja käytettyjen palvelinohjelmistojen tehokkuus. Tässä yhteydessä on syytä huomauttaa, että kappaleessa 7.4 esitelty yleinen suorituskykymalli on yleinen tulostinmalli, ei yleinen järjestelmämalli. Tämä tarkoittaa käytännössä, että koska syöteinä on annettu ainoastaan tulostimia kuvaavia parametreja, voidaan mallilla ennustaa ainoastaan sellaisten tulostimien suorituskykyä, jotka on tarkoitus liittää osaksi opetusaineiston tulostamiseen käytettyä järjestelmää. Verkon suorituskyky luokitellaan siis tässä tapauksessa kiinteäksi osaksi malleja.



## 8.1.4 Yleistä kehitetyistä algoritmeista

Jäljempänä esitetyistä hajautettujen painatusjärjestelmien arkkitehtuurikuvauksista tul- laan havaitsemaan, että hajautetun järjestelmän mallintaminen yhden ainoan kaavan avulla on mahdotonta. Vaihtoehtoisia tiedosto- ja järjestelmäkombinaatioita on niin run- saasti, että ongelma on pakko jakaa pienemmiksi osiksi jotta sen ratkaiseminen olisi mah- dollista. Ainoa järkevä tapa toteuttaa tämän tyyppinen malli on koodata useat kokonaisuuden kuvaamiseen tarvittavat algoritmit jollakin ohjelmointikielellä, jolloin tapauskohtaiset piirteet pystytään huomioimaan ennusteita laskettaessa. Algoritmit pää- dyttiin koodaamaan C-kielellä ja lähdekoodi on esitetty liitteessä 6. Ohjelman rakenne on hyvin yksinkertainen. Ensimmäiseksi valitaan haluttu tulostusarkkitehtuuri seuraavista vaihtoehdoista:

- Useampia tiedostoja tulostetaan peräkkäin, ilman viivettä yhdelle tulostimelle eli tavoitteena oli simuloida "pipeline"-tyyppistä tulostettavan materiaalin käsittelyä.
- Yhdeltä työasemalta tulostetaan useammalle tulostimelle.
- Useammalta työasemalta tulostetaan samanaikaisesti useammalle tulostimelle.

Ohjelmaa kehitettäessä on siihen liitetty muutama algoritmien testaamista helpottava omi- naisuus, joiden avulla voidaan hieman nopeuttaa sen käyttämistä. Seuraavassa esitetyt ominaisuudet ovat kaikille yllä kuvatuille arkkitehtuurimalleille yhteisiä. Ensimmäinen huomioitu seikka on tulostettavan materiaalin rakenne eli muodostuuko se useammasta kappaleesta samaa tiedostoa vai poikkeako tulostettavien tiedostojen sisältö toisistaan. Jos tulostetaan useita kappaleita yhdestä ainoasta tiedostosta, tarvitsee sitä kuvaavat para- metrit (tiedoston koko ja arvioitu tulostusaika) syöttää ainoastaan kerran eli näin vältetään syöttämästä samaa informaatiota useita kertoja peräkkäin. Jos taas tulostetaan useita toi- sistaan poikkeavia tiedostoja, joudutaan edellä mainitut kaksi parametriä syöttämään erik- seen jokaisen tiedoston kohdalla. Käytettävästä työasemasta riippuva tiedonsiirtonopeus huomioidaan algoritmeissa vakiona. Kaikkien käytettyjen työasemien tiedonsiirtonopeudet on syötetty ohjelmaan valmiiksi ja käyttäjältä kysytään ainoastaan miltä työasemalta tie- dostoa ollaan lähettämässä. Jos kaikki tiedostot lähetetään samalta työasemalta, kysytään tietoa ainoastaan kerran. Ennustettaessa useammasta työasemasta muodostuvan järjestel- män suorituskykyä, joudutaan lähettävä asema valitsemaan erikseen jokaisen lähetettävän tiedoston kohdalla.

Varsinaisten suorituskykyalgoritmien tarkempi kuvaus on esitetty samassa yhteydessä kuin mallinnettava tulostusarkkitehtuuri on analysoitu. Näiden C-koodattujen algoritmien osalta kannattaa kuitenkin huomata, että niiden ympärille rakennettu käyttöliittymä on hyvin alkeellinen ja suhteellisen kömpelö käyttää, eikä sitä ole tarkoitettu tässä muodossa käytännössä sovellettavaksi. Käyttäjä joutuu syöttämään manuaalisesti lähes kaikki tarvit- tavat muuttujat, koska mitään kytkentää varsinaiseen tulostusjärjestelmään ei ole toteu- tettu. Todellisuudessa suurin osa näistä käyttäjän kannalta kömpelöistä operaatioista voidaan hoitaa automaattisesti kun algoritmit on integroitu osaksi tulostusjärjestelmää. Tämän diplomityön yhteydessä ohjelmoitu sovellus on siis tarkoitettu ainoastaan algori- tmien toimivuuden tutkimiseen ja todistamiseen.

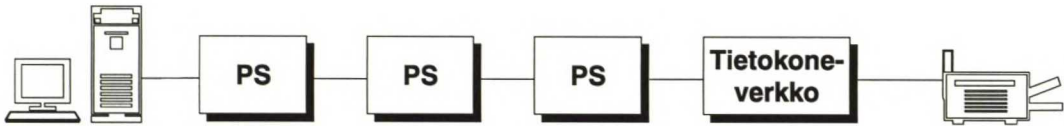
## 8.2 "Pipeline"-simulaatio

### 8.2.1 Järjestelmäanalyysi

Kuvassa 8.2.2 on esitetty koejärjestely, jolla pyrittiin simuloimaan niin sanottua "pipeline"- tulostusarkkitehtuuria. Kyseessä ei kuitenkaan ole aito "pipeline"-järjestelmä, koska tulos-



timella kukin toimenpide suoritetaan vuorollaan, eikä niiden välillä ole päällekkäisyyttä. Aidoissa "pipeline"-järjestelmissä itse tulostin pystyy suorittamaan useampia toimintoja samanaikaisesti. Kuvassa esitetyn kaltaista järjestelmää voidaan kuitenkin siinä mielessä pitää "pipeline"-toteutuksena, että useampia tiedostoja tulostettaessa seuraavaa tiedostoa voidaan ryhtyä siirtämään tulostusjonoon välittömästi sen jälkeen, kun edellisen tiedoston siirto on päättynyt ja sen käsittely tulostimella on aloitettu. Ensimmäisen tiedoston jälkeen tulostettavien tiedostojen kohdalla voitetaan siis tiedoston siirtoon työasemalta tulostusjonoon kulunut aika. Juuri tästä syystä on tärkeää, että siirtoon kulunut aika pystytään erottamaan ennustetusta kokonaistulostusajasta. Kokonaistulostusaika muodostuu siis lyhyemmäksi kuin mitä erillisinä tulostettujen tiedostojen yhteenlasketut tulostusajat. Saa-vutettu hyöty on sitä suurempi, mitä useampia tiedostoja tulostetaan.



**Kuva 28.** "Pipeline"-simulaatiossa hyödynnetty koeympäristö. Tämän tyyppinen järjestelmä poikkeaa kuvassa 27 esitetystä siinä mielessä, että PS-tiedostot lähetetään peräkkäin ilman viivettä.

Edellä esitetty konsepti pätee kuitenkin ainoastaan siinä tapauksessa, että tulostettavan tiedoston tulostusaika on pidempi kuin seuraavan tiedoston siirtämiseen kuluva aika. Jos edellinen tiedosto ehditään tulostaa valmiiksi ennenkuin seuraava on kokonaan siirretty tulostusjonoon, ei koko siirtoaikaa voida vähentää tiedoston ennustetusta kokonaistulostusajasta. Tällöin voitettu aika on ainoastaan edellisen tiedoston tulostamiseen kulunut aika. Siinä tapauksessa, että kaikki tulostettavat tiedostot ovat sisällöltään samanlaisia kyseistä ongelmaa ei synny, koska tiedostojen tulostaminen kestää aina kauemmin kuin siirto työasemalta tulostusjonoon. Jos taas tulostetaan useita sisällöltään poikkeavia tiedostoja, saat- taan syntyä tilanne, jossa seuraavan tiedoston siirto on vielä kesken kun edellinen on jo tulostettu.

Mallissa on lisäksi huomioitava yksi tulostinkohtainen ominaisuus. Kun tulostettavan työn viimeinen sivu on rasteroitu valmiiksi, se siirretään fotojohteelta luovutuskaukaloon. Koska yhden tulostimen mallinnuksessa kaikki tiedostot tulostetaan täysin erillisinä ja tulostuk- sen katsotaan päättyneen vasta kun viimeinen sivu on luovutuskaukalossa, tulee tämä arkin siirtämiseen kulunut aika automaattisesti huomioitua jokaisen tiedoston tulostus- ajassa ja siten myös mallien avulla lasketuissa ennusteissa. Jos seuraava tulostettava tie- dosto on jo valmiina palvelimen tulostusjonossa odottamassa tulostusvuoroaan, voidaan sitä tämän paperiarkin siirron aikana ryhtyä siirtämään palvelimen tulostusjonosta tulostimen työmuistiin. Tässä tapauksessa fotojohteelta luovutuskaukaloon siirtämiseen kulunut aika täytyy vähentää ensimmäistä tiedostoa seuraavien tiedostojen arvioiduista tulostusajoista, koska toiminnot ovat päällekkäisiä.

Testitulostuksissa kävi ilmi, että tämä siirtoaika fotojohteelta luovutuskaukaloon on kulla- kin tulostimella lähes vakio eli se voidaan huomioida mallissa vastaavalla tavalla kuin käy- tetyn työaseman datasiirtonopeus. Koska arkin siirtoon kuluvan ajan mittaaminen on kuitenkin suhteellisen hankalaa siitä syystä, että rasteroinnin päättymishetki on vaikea havaita, määritettiin se laskennallisesti. Teoriassa voidaan olettaa, että tulostettaessa use- ampia samoja tiedostoja, kokonaistulostusaika muodostuu seuraavista osista:

- ensimmäisen tulostettavan tiedoston työasemalta palvelimelle siirtämiseen kulu- vasta ajasta ja
- kaikkien tulostettavien tiedostojen tulostusajoista, joissa ei siis huomioitu siirtoai- kaa työasemalta palvelimelle.



Todellisuudessa kokonaistulostusaika on kuitenkin aina lyhyempi kuin näin määritetty aika. Voidaankin olettaa, että juuri tämä erotus on aika, joka säästetään johtuen toimintojen päällekkäisyydestä tulostimella. Fotojohteelta luovutuskaukaloon siirtämiseen kuluva aika voidaan siis määrittää jakamalla tämä erotus ensimmäistä tulostettavaa tiedostoa seuraavien tiedostojen lukumäärällä. Tähän perustuu myös väite, että mitä useampia tiedostoja tulostetaan, sitä suurempi on saavutettu suhteellinen ajansäästö verrattuna erikseen tulostettujen tiedostojen yhteenlaskettuun kokonaistulostusaikaan. Edellä kuvatulla tavalla lasketut paperiarkin siirtämiseen kuluvat ajat kunkin testitulostimen osalta ovat

- |                              |          |
|------------------------------|----------|
| • HP LaserJet 4/4M+ PS       | 14,80 s. |
| • IBM LaserPrinter 4029 PS17 | 22,99 s. |
| • Canon CLC500+Fiery 125i    | 4 s.     |

Kuten huomataan Canonin kohdalla aika on selvästi lyhyempi kuin kahden muun tulostimen tapauksessa. Tämä johtuu Canonin ja Fiery RIPin toteutustavasta eli rasterointiprosessori ei aloita seuraavan työn käsittelyä, ennen kuin edellinen työ on täysin valmiina luovutuskaukalossa. Tällä perusteella ajan pitäisi kuitenkin olla nolla sekuntia. Syy tähän löytyy luultavasti siitä, että jos seuraava tulostettava tiedosto on edellisen tiedoston tulostuksen päättyessä tallennettu Fiery RIPin kovalevylle, se on ikäänkuin valmiiksi tulostimen työmuistissa. Näin osa tulostuksen aloittamisen vaatimista valmisteluista on jo tehty. Niitä ei tarvitse enää rasterointia aloitettaessa suorittaa ja näin säästetään mainitut neljä sekuntia. Yllä esitetyt ajat saattavat tuntua vähäpätöisiltä, mutta koska niiden määrittäminen on suhteellisen vaivatonta kannattaa se mallinnusvaiheessa tehdä, koska näin varmistetaan tarkemmat ennusteet. Lisäksi kannattaa muistaa, että kun tulostettavien tiedostojen lukumäärä kasvaa, myös kyseisten aikojen vaikutus kasvaa samassa suhteessa.

## 8.2.2 Algoritmien kuvaus

Edellisen kappaleen järjestelmäanalyysin perusteella laaditut algoritmit on esitetty C-koodattuna liitteessä 6 aliohjelmassa **one\_to\_one**. Seuraavassa on yksityiskohtaisesti kuvattu aliohjelman toiminta. Ensimmäisessä vaiheessa selvitetään tulostettavan materiaalin rakenne eli muodostuuko se useasta samasta tiedostosta vai poikkeavatko tulostettavat tiedostot sisällöltään toisistaan. Tämän jälkeen valitaan työasema ja tulostin. Näiden tietojen perusteella määrätään kaavoissa käytettävä työaseman ja palvelimen välinen datasiirtonopeus, sekä tulostimen arkinsiirtonopeus. Tässä yhteydessä selvitetään myös, kuinka monta kopiota yhdestä tiedostosta halutaan tai kuinka monta erilaista tiedostoa halutaan tulostaa. Tämän jälkeen ohjelma jakaantuu kahteen osaan riippuen ensimmäisestä valinnasta eli tulostetaanko useita kappaleita samasta tiedostosta vai useita eri tiedostoja.

Siinä tapauksessa, että ollaan tulostamassa useita kappaleita samasta tiedostosta selvitetään ainoastaan arvioitu tulostusaika valitulla tulostimella ja tulostettavan tiedoston koko. Tämän jälkeen lasketaan tiedoston siirtoon kuluva aika jakamalla tiedoston koko valitun työaseman datasiirtonopeudella. Lopullinen kokonaistulostusaika lasketaan siirtoajasta ja arvioidusta tiedoston tulostusajasta seuraavalla kaavalla:

$$\begin{aligned} \text{kokonaistulostusaika} = & \text{ensimmäisen tiedoston siirtoaika} + \\ & (\text{tiedostojen lukumäärä} \times \text{arvioitu tulostusaika}) - \\ & ((\text{tiedostojen lukumäärä} - 1) \times \text{paperiarkin siirtoaika}) \end{aligned} \quad (5)$$

Jos sen sijaan halutaan tulostaa useampia erilaisia tiedostoja, käytetään periaatteessa samaa kaavaa, mutta hieman monimutkaisemmassa muodossa. Koska tulostettavat tiedostot poikkeavat ominaisuuksiltaan toisistaan, joudutaan jokainen näistä tiedostoista käsittelemään ikäänkuin omana erillisenä kokonaisuutenaan. Yksitellen selvitetään kunkin tulostettavan tiedoston osalta arvioitu tulostusaika ja tiedoston koko. Kun tiedetään, miltä



työasemalta tiedosto lähetetään voidaan laskea työasemalta palvelimelle siirtämiseen kuluva aika. Jos kyseessä on ensimmäinen tulostettava tiedosto, lasketaan sen kokonaistulostusaika summaamalla siirtoon kulunut aika arvioidun tulostusajan kanssa. Tämän jälkeen selvitetään seuraavan tiedoston arvioitu tulostusaika ja tiedoston koko, sekä lasketaan siirtoon kuluva aika. Tämän tiedoston siirtoaikaa verrataan edellisen tiedoston tulostusaikaan: jos tiedoston siirtäminen kestää kauemmin kuin edellisen tulostaminen, lasketaan tämän tiedoston kokonaistulostusaika summaamalla arvioituun tulostusaikaan siirtoaika, josta on vähennetty edellisen tiedoston arvioitu tulostusaika. Jos taas siirto työasemalta palvelimelle kestää vähemmän aikaa kuin edellisen tulostaminen, muodostuu kokonaistulostusajaksi arvioitu tulostusaika, josta on nyt vähennettävä tulostinkohtainen paperiarkin siirtämiseen kuluva aika. Tätä operaatiota toistetaan, kunnes kaikki tulostettavat tiedostot on käyty läpi. Näin määritetyt kunkin tiedoston kokonaistulostusajat lasketaan yhteen ja saadaan lopullinen kaikkien tiedostojen yhteinen tulostusaika.

8.2.3 Kehitettyjen algoritmien testaus

Kuten aiemmin mainittiin, käytettiin kehitettyjen mallien testauksessa samaa materiaalia kuin neuroverkkomallien opetuksessa. Liitteessä 5 taulukoissa 1, 2 ja 3 on esitetty kunkin testitulostimen osalta tässä kappaleessa kehitettyjen mallien testauksessa käytettyjen testisarjojen rakenne eli montako ja minkä tyyppisiä tiedostoja ne sisälsivät. Kussakin taulukossa on lisäksi esitetty tulostusvaiheessa mitatut kokonaissiirtoajat ja niiden perusteella määritellyt tiedostokohtaiset siirtoajat. Näiden aikojen ja tiedostokoon perusteella on laskettu keskimääräiset tiedostokohtaiset siirtonopeudet [kt/s] käytetyltä työasemalta (No. 1). Lisäksi on vielä taulukoitu kunkin testisarjan lopullinen mitattu tulostusaika, jota verrataan mallien avulla ennustettuihin tulostusaikoihin jäljempänä tässä kappaleessa esitetyissä taulukoissa. Tässä kappaleessa on taulukoissa 10, 11 ja 12 esitetty kunkin testitulostimen osalta liitteessä kuvattujen testisarjojen mitatut, sekä mallien avulla ennustetut tulostusajat. Näiden tulostusajojen välisestä erosta on laskettu virheprosentti kaavalla 3. Näistä virheprosentteista on edelleen laskettu mallien tekemä keskimääräinen virhe, virheiden otoskeskihajonta ja otosvarianssi täsmälleen samalla tavalla kuin aiempien mallien kohdalla.

Taulukko 10. HP LaserJet 4/4M+ PS-tulostimen "pipeline"-simulaatiossa mitattujen ja ennustettujen tulostusajojen vertailu, sekä niiden välisestä erotuksesta lasketut virheprosentit.

Testisarjan no. (HP)	Todellinen tulostusaika [s]	Todellinen tulostusaika [min:s]	Ennustettu tulostusaika [s]	Ennustettu tulostusaika [min:s]	Virhe [%]
I	790	13:10	782	13:02	1,01 %
II	2149	35:49	2161	36:01	0,56 %
III	1669	27:49	1673	27:53	0,24 %
IV	556	09:16	555	09:15	0,18 %
V	194	03:14	192	03:12	1,03 %
Keskimääräinen virhe					0,60 %
Virheen otoskeskihajonta					0,41 %
Virheen otosvarianssi					0,00 %



**Taulukko 11.** IBM LaserPrinter 4029 PS17-tulostimen "pipeline"-simulaatiossa mitattujen ja ennustettujen tulostusaikojen vertailu, sekä niiden välisestä erotuksesta lasketut virheprosentit.

Testisarjan no. (IBM)	Todellinen tulostusaika [s]	Todellinen tulostusaika [min:s]	Ennustettu tulostusaika [s]	Ennustettu tulostusaika [min:s]	Virhe [%]
I	1976	32:56	1963	32:43	0,66 %
II	5551	92:31	5619	93:39	1,23 %
III	3969	66:09	3973	66:13	0,10 %
IV	1276	21:16	1267	21:07	0,71 %
V	635	10:35	637	10:37	0,31 %
Keskimääräinen virhe					0,60 %
Virheen otoskeskihajonta					0,43 %
Virheen otosvarianssi					0,00 %

**Taulukko 12.** Canon CLC 500+Fiery 125i-tulostimen "pipeline"-simulaatiossa mitattujen ja ennustettujen tulostusaikojen vertailu, sekä niiden välisestä erotuksesta lasketut virheprosentit.

Testisarjan no. (Canon)	Todellinen tulostusaika [s]	Todellinen tulostusaika [min:s]	Ennustettu tulostusaika [s]	Ennustettu tulostusaika [min:s]	Virhe [%]
I	825	13:45	829	13:49	0,48 %
II	2800	46:40	2841	47:21	1,46 %
III	2796	46:36	2837	47:17	1,47 %
IV	1959	32:39	1963	32:43	0,20 %
V	604	10:04	599	09:59	0,83 %
Keskimääräinen virhe					0,89 %
Virheen otoskeskihajonta					0,57 %
Virheen otosvarianssi					0,00 %

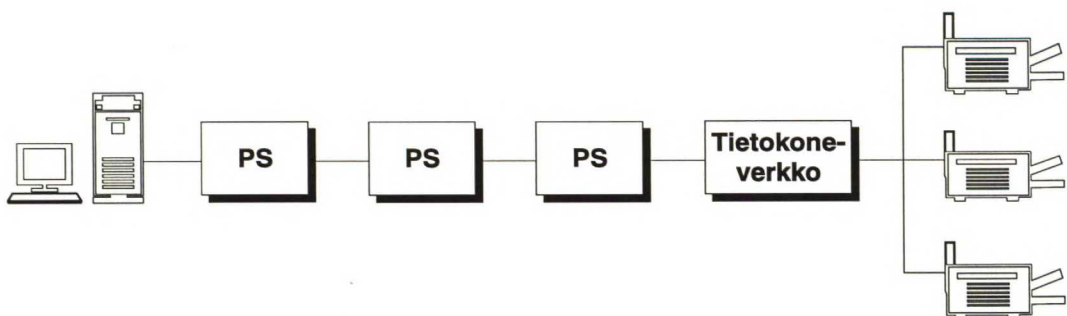
Kun tarkastellaan taulukoissa esitettyjä virheprosentteja, voidaan malleja pitää erittäin tarkkoina. Keskimääräiset virheet ovat kaikkien kolmen tulostimen tapauksessa selvästi alle prosentin luokkaa ja otoskeskihajonnat ja otosvarianssit ovat vielä pienempiä. Kehitettyjä malleja voidaan pitää ainakin testijärjestelmän kaltaisessa ympäristössä täysin luotettavina ja suhteellisen turvallinen oletus on myös, että sovellettaessa niitä muihin ympäristöihin niiden tarkkuus on myös tuolloin riittävä. Kannattaa kuitenkin pitää mielessä, että näin tarkat tulokset edellyttävät yksittäisten tiedostojen tulostusaikojen erittäin tarkkaa ennustamista ja siihen ei vielä tässä työssä kehitetyillä neuroverkkomalleilla päästy. Jos edellä kuvatuissa testeissä olisi käytetty näiden mallien avulla ennustettuja yhden tulostimen tulostusaikoja, olisivat virheprosentit olleet luultavimmin muutamaa keraluokkaa suurempia.

## 8.3 Yhden työaseman ja useamman tulostimen ympäristöt

### 8.3.1 Järjestelmäanalyysi

Kuvassa 29 on esitetty edellisessä kappaleessa rakennettua järjestelmää hieman monimutkaisempi arkkitehtuuri. Tässä ympäristössä tulostetaan yhdeltä työasemalta useammalle tulostimelle eli teoriassa voidaan puhua rinnakkaislaskennan hyödyntämisestä tulostettavaa materiaalia käsiteltäessä. Tässä työssä tulostimien lukumäärä on aina sama kuin tulostettavien tiedostojen lukumäärä. Tämä ei ole välttämätöntä, mutta algoritmi haluttiin pitää mahdollisimman yksinkertaisena tulosten tulkinnan helpottamiseksi. Kehitettyä algoritmia on kuitenkin helppo tarvittaessa laajentaa käsittelemään tapauksia, joissa tulostimille lähetetään useampia tiedostoja. Tällöin ainoastaan yhdistetään edellisen kappaleen "pipeline"-algoritmit tässä kappaleessa kehitettyihin algoritmeihin.

Järjestelmä ei sinänsä ole reaalisesti rinnakkainen, koska lähetettäessä tiedostoja työasemalta käsitellään niitä edelleen ikäänkuin sarjassa eli ainoastaan yksi tiedosto pystytään lähettämään kerrallaan ja muut tulostettavat tiedostot joutuvat odottamaan käsittelyvuoroaan. Sen jälkeen kun kaikki tiedostot on saatu sijoitettua palvelimen tulostusjonoihin, käsittely voidaan kuitenkin pitää periaatteessa rinnakkaisena. Tämän jälkeen suorituskykyyn vaikuttaa ainoastaan palvelimen ja tulostimen välisen tiedonsiirron tehokkuus, sekä itse tulostimen suorituskyky. Nimenomaan nämä järjestelmäominaisuudet on otettu huomioon yksittäisen tulostimen suorituskykyä ennustettaessa, joten niitä ei enää tässä yhteydessä tarvitse käsitellä mitenkään. Kunkin tiedoston arvioitua tulostusaikaa voidaan näin ollen käyttää sellaisenaan kokonaissuorituskykyä laskettaessa.



**Kuva 29.** Tulostusjärjestelmäarkkitehtuuri, joka muodostuu yhdestä lähettävästä työasemasta ja useammasta tulostimesta.

Vaikka arkkitehtuuri on periaatteessa monimutkaisempi kuin edellisessä kappaleessa esitetty yhden työaseman ja yhden tulostimen ympäristö, on sen mallintaminen kuitenkin yksinkertaisempaa. Jokaiselle käytettävälle tulostimelle lähetetään ainoastaan yksi tiedosto ja tulostimien välillä ei mahdollista yhteistä tiedonsiirtomediaa lukuunottamatta ole muita riippuvuuksia. Näin ollen tulostimen koneistonopeudesta riippuvia muuttujia ei tarvitse mallinnuksessa huomioida. Tämä tarkoittaa käytännössä, että suorituskyvyn kannalta merkittäviä tekijöitä ovat ainoastaan työasemalta palvelimelle siirtoon kulunut aika ja itse tulostukseen kulunut aika. Koska tiedostot lähetetään palvelimelle peräkkäin, muodostuu kokonaistulostusaika kunkin tiedoston osalta edellä mainituista siirto- ja tulostusajoista joihin on lisätty aika, joka joudutaan odottamaan edellisten tulostettavien tiedostojen siirtämistä palvelimelle. Tässä vaiheessa mahdollinen muu verkkokuorma voi



vaikuttaa muodostuviin siirtoaikoihin, mutta sen huomioiminen on mallissa erittäin vaikeaa. Lopulliseksi, koko järjestelmän suorituskykyä kuvaavaksi tulostusajaksi muodostuu siis sen tiedoston tulostusaika, joka on edellä kuvatulla tavalla laskettuna pisin.

Jokaisen tässä työssä käytetyn tulostimen tapauksessa palvelimen ja tulostimien välinen yhteys on toteutettu eri tavalla eli muodostetun mallin voidaan katsoa soveltuvan tulostusaikojen ennustamiseen riippumatta siitä, miten palvelimen ja tulostimen välinen tiedonsiirto tapahtuu. Huomattavaa on, että jos tulostimien määrä kasvaa riittävän suureksi ja tiedostoja siirrettäessä hyödynnetään yhteistä siirtomediaa, saattaa sen siirtokapasiteetti muodostua pullonkaulaksi ja tällöin tulostusjärjestelmän rinnakkaisprosessoinnin kautta saavutettu hyöty ei enää ole yhtä merkittävä. Tässä tapauksessa kyseiseen ongelmaan ei kuitenkaan törmätty, koska käytettiin ainoastaan kolmea tulostinta. Laajemmissa järjestelmissä saattaa olla tarpeen huomioida tästä verkkokuormasta aiheutuva viive kokonaistulostusaikoja laskettaessa.

### 8.3.2 Algoritmien kuvaus

Edellä kuvattu malli on esitetty C-koodattuna liitteessä 6 aliohjelmassa **one\_to\_N**. Vastavalla tavalla kuin edellisessä kappaleessa kuvatussa aliohjelmassa, myös tässä tapauksessa selvitetään ensimmäiseksi muodostuuko tulostettava materiaali useasta samansisältöisestä vai useista toisistaan poikkeavista tiedostoista. Edelleen selvitetään käytettävä työasema, jotta pystytään määrittämään sen ja palvelimen välinen tiedonsiirtonopeus. Samassa yhteydessä selvitetään lisäksi käytettävien tulostimien lukumäärä, joka on samalla tulostettavien tiedostojen lukumäärä. Tämän jälkeen aliohjelma on jaettu kahteen osaan riippuen siitä, tulostetaanko useita kappaleita samasta tiedostosta vai useita eri tiedostoja. Nämä osat eroavat toisistaan ainoastaan tiedoston koon selvittämisen osalta. Jos tulostetaan useita samansisältöisiä tiedostoja, selvitetään tiedoston koko ainoastaan kerran, muussa tapauksessa se joudutaan selvittämään jokaisen tiedoston kohdalla erillisenä työvaiheena. Koska algoritmit ovat hyvin samankaltaisia, niiden peruseräite on tässä kappaleessa kuvattu yhtenä kokonaisuutena, vaikka ne todellisuudessa on toteutettu erillisinä aliohjelmina.

Edellä mainittujen tietojen lisäksi on vielä selvittettävä arvioitu tulostusaika kunkin tiedoston osalta. Tässä tapauksessa se on tehtävä erillisenä työvaiheena tiedostokohtaisesti, koska jokainen tiedosto tulostetaan eri tulostimella ja näiden suorituskyky poikkeaa merkittävästi toisistaan. Algoritmia voidaan kehittää edelleen siten, että jos järjestelmässä on useampia ominaisuuksiltaan samankaltaisia tulostimia, niitä koskevat muuttujat voidaan selvittää samanaikaisesti yhtenä työvaiheena. Tämän jälkeen lasketaan työasemalta palvelimelle kuluva siirtoaika työaseman suorituskyvyn ja lähetettävän tiedoston koon perusteella. Jos käsiteltävänä on ensimmäinen lähetettävä tiedosto, voidaan kokonaistulostusajaksi asettaa suoraan lasketun siirtoajan ja arvioidun tulostusajan summa. Samalla tallennetaan muistiin kokonaissiirtoaika, jota kasvatetaan kunkin lähetettävän tiedoston osalta siirtoon kuluneella ajalla.

Tämän jälkeen kunkin tulostettavan tiedoston osalta siirto- ja tulostusajan summaan lisätään aiemmin lähetettyjen tiedostojen yhteenlasketut siirtoajat. Näin saadaan laskettua kullekin tulostettavalle tiedostolle lopullinen tulostusaika. Koska tulostettavien tiedostojen välillä ei palvelimelle lähettämisen jälkeen ole merkittäviä riippuvuuksia, voidaan tulostusjärjestelmän kokonaistulostusajaksi määritellä tiedostokohtaisista tulostusajoista pisin. Tämä tarkoittaa käytännössä, että tiedosto jonka tulostaminen edellä kuvatulla tavalla laskettuna kestää pisimpään, määrää koko järjestelmän suorituskyvyn (kokonaistulostusajan). Ohjelmassa tämä on toteutettu yksinkertaisella silmukkarakenteella, jossa kunkin tulostettavan tiedoston laskettua tulostusaikaa verrataan siihen mennessä pisimpään tulostusaikaan. Jos viimeisin aika on pidempi kuin aiemmin laskettu, sijoitetaan tämä uusi tulostusaika koko järjestelmän suorituskykyä kuvaavaksi tulostusajaksi.



8.3.3 Kehitettyjen algoritmien testaus

Taulukossa 13 on esitetty todelliset ja edellä kuvatuilla algoritmeilla ennustetut tulostusajat. Kunkin testisarjan rakenne on kuvattu liitteessä 5 taulukossa 4. Liitteen taulukosta käy ilmi kunkin testisarjan rakenne eli tulostetut tiedostot, niiden lukumäärä ja millä tulostimella kukin niistä on tulostettu. Lisäksi taulukkoon on koottu kellotetut siirtoajat ja niiden perusteella lasketut tiedostokohtaiset siirtoajat. Myös näissä testeissä käytettiin edelleen ainoastaan työasemaa numero 1 eli tiedostokokojen ja siirtoaikojen perusteella laskettuja siirtonopeuksia käytettiin jo aiemmin laskettaessa työaseman 1 keskimääräistä siirtonopeutta. Lopuksi on vielä esitetty kunkin testisarjan kokonaistulostusaika. Tämä aika on määritelty alkavaksi sillä hetkellä, kun ensimmäistä tulostettavaa tiedostoa ryhdytään siirtämään ja päättyväksi sillä hetkellä, kun viimeinen sivu siitä tiedostosta, jonka tulostaminen kestää pisimpään on tulostimen luovutuskaukalossa.

Taulukko 13. Yhden työaseman ja useamman tulostimen ympäristössä mitatut ja kehitetyillä malleilla ennustetut tulostusajat.

Testisarjan no.	Todellinen tulostusaika [s]	Todellinen tulostusaika [min:s]	Ennustettu tulostusaika [s]	Ennustettu tulostusaika [min:s]	Virhe [%]
I	135	02:15	137	02:17	1,48 %
II	1081	18:01	1072	17:52	0,83 %
III	800	13:20	805	13:25	0,63 %
IV	412	06:52	411	06:51	0,24 %
V	5072	84:32	5077	84:37	0,10 %
VI	1633	27:13	1645	27:25	0,73 %
VII	774	12:54	768	12:48	0,78 %
VIII	735	12:15	722	12:02	1,77 %
IX	275	04:35	274	04:34	0,36 %
X	309	05:09	301	05:01	2,59 %
Keskimääräinen virhe					0,95 %
Virheen otoskeskihajonta					0,77 %
Virheen otosvarianssi					0,01 %

Kuten yllä olevasta taulukosta havaitaan, ovat virheprosentit todella pieniä eli keskimäärin alle prosentti luokkaa. Myös virheiden otoskeskihajonta ja otosvarianssi ovat selvästi alle prosentti luokkaa. Tästä voidaan päätellä, että kuten edellisten "pipeline"-mallien tapauksessa, myös tässä arkkitehtuurissa tulostusjärjestelmän sisäiset riippuvuudet ovat riittäväällä tarkkuudella lineaarisia ja kokonaissuorituskykyä voidaan ennustaa edellä kuvattujen yksinkertaisten kaavojen avulla. Jälleen on huomattava, että näin tarkkojen kokonaistulostusaikaennusteiden edellytyksenä on riittävän tarkat yksittäisten tulostimien tulostusaikaennusteet. Tehdyt testit todistavat siis ainoastaan, että järjestelmää kuvaavat matemaattiset mallit toimivat. Se miten tarkasti todellisuudessa pystytään järjestelmän suorituskykyä ennustamaan riippuu täysin siitä, kuinka tarkoiksi yksittäisten tulostimien suorituskyvyn ennustamiseen käytetyt neuroverkkomallit voidaan opettaa.

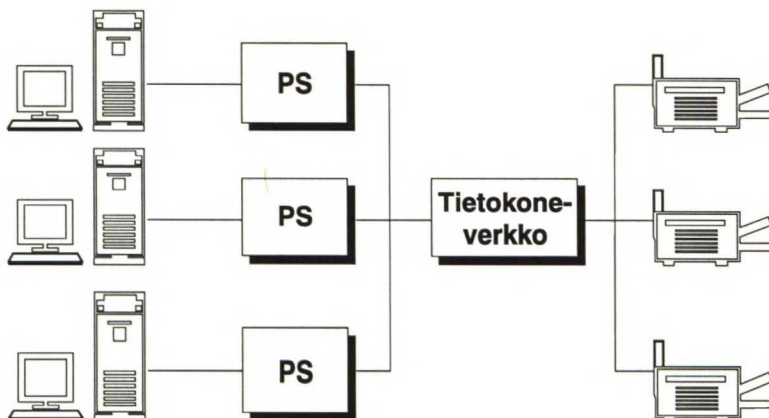


Lisäksi on huomattava, että tässä tapauksessa verkkokuorman voidaan olettaa näyttävän huomattavasti merkittävämpää roolia kuin "pipeline"-simulaatioiden tapauksessa. "Pipeline"-simulaatiossa kukin tiedosto lähetettiin vuorollaan palvelimelle ja sieltä edelleen tulostimelle. Näin ollen siirtomediaa hyödynnetään samanaikaisesti ainoastaan silloin, kun edellistä tiedostoa siirretään palvelimelta tulostimelle ja seuraavaa työasemalta palvelimelle. Tällöin kuormitus on kuitenkin niin alhainen, että sen vaikutus voidaan aina ilman muuta jättää huomiotta. Yhden työaseman ja useamman tulostimen ympäristössä sen sijaan on mahdollista, että tiedostojen siirto työasemalta palvelimelle ja sieltä edelleen useammalle tulostimelle tapahtuu samanaikaisesti eli käytettävän siirtomedian kuormitus kasvaa merkittävästi edellä kuvattuun tilanteeseen verrattuna. Kuten aiemmin jo todettiin, koeympäristössä verkkokuormitus ei vielä vaikuttanut merkittävästi, mutta laajemmissa ympäristöissä sen huomioiminen voi olla tarpeellista.

## 8.4 Useamman työaseman ja useamman tulostimen ympäristöt

### 8.4.1 Järjestelmäanalyysi

Kuvassa 30 on esitetty viimeinen ja rakenteeltaan monimutkaisin tässä työssä käsitellyistä tulostusarkkitehtuureista. Siinä tulostetaan useammalta työasemalta samanaikaisesti useammalle tulostimelle. Nyt järjestelmää voidaan teoriassa pitää täysin rinnakkaisena, koska lähetettävien tiedostojen ei enää tarvitse odottaa lähetysvuoroaan kuten edellisen arkkitehtuurin tapauksessa. Käytännössä on otettava huomioon, että koska tiedostojen siirtoon käytetään samaa siirtomediaa, voidaan ajautua tilanteeseen jossa verkko menee kuormituksen takia tukkoon. Tämä edellyttää kuitenkin, että järjestelmä sisältää runsaasti sekä lähetäviä työasemia, että tulostimia. Toinen vaihtoehto on, että verkossa on runsaasti muuta liikennettä. Tämän tyyppisen järjestelmän mallintamisen perustana on, että tunnetaan kaikkien käytettävien työasemien suorituskyky ja lisäksi on tunnettava kunkin käytetyn tulostimen suorituskyky. Työasemien mallintaminen on helppoa testitulostusten avulla ja tulostimien suorituskykyä pyritään ennustamaan aiemmin käsiteltyjen neuroverkkomallien avulla. Kuten edellisen arkkitehtuurin tapauksessa, myös tässä ympäristössä mallintaminen on itseasiassa yksinkertaisempaa kuin aluksi voisi olettaa.



**Kuva 30.** Tulostusjärjestelmämalli, jossa useammalta työasemalta lähetään samanaikaisesti tiedostoja useammalle tulostimelle.



Algoritmien testaamisen helpottamiseksi myös tässä tapauksessa työasemien ja tulostimien lukumäärän on oletettu olevan sama, eli esimerkiksi arkkitehtuuria jossa kahdelta työasemalta lähetetään tiedostoja kolmelle tulostimelle ei ole testattu. Tämän tyyppisen mallin rakentaminen on kuitenkin muodostetun perusalgoritmin pohjalta helppoa. Kuvatussa ympäristössä sekä tulostettavan tiedoston lähettäminen palvelimelle, että sen lähettäminen edelleen valitulle tulostimelle on periaatteessa muista järjestelmän toiminnoista riippumattonta. Tästä syystä voidaan tehdä yksinkertainen oletus, että järjestelmän kautta tulostettavien tiedostojen kokonaistulostusajaksi muodostuu sen tiedoston tulostusaika, jonka tulostaminen kestää pisimpään. Mallintaminen voidaan siis tehdä siten, että lasketaan kullekin tulostettavalle tiedostolle arvioitu tulostusaika huomioiden käytettävän työaseman ja kohdetulostimen ominaisuudet. Teoriassa kunkin tulostustapahtuman voidaan katsoa olevan muista samanaikaisista tulostusoperaatioista riippumattomia. Koko järjestelmän suorituskykyä kuvaa siis sen tiedoston tulostusaika, jonka tulostaminen kestää pisimpään.

## 8.4.2 Algoritmien kuvaus

Edellä kuvatun teorian pohjalta laaditut C-koodatut algoritmit on esitetty liitteessä 6 aliohjelmassa **N\_to\_N**. Kuten kahden edellisen aliohjelman tapauksessa, myös tässä algoritmissa selvitetään ensimmäiseksi tulostettavan materiaalin rakenne. Sen jälkeen selvitetään lähetettävien työasemien lukumäärä, joka on sama kuin käytettyjen tulostimien lukumäärä. Näiden tietojen selvittämisen jälkeen ohjelma jakaantuu kahteen osaan riippuen siitä, muodostuuko tulostettava materiaali useasta samasta tiedostosta vai useista sisällöltään poikkeavista tiedostoista. Tämä jako on ainoastaan periaatteellinen ja sen ainoa tarkoitus on hieman helpottaa ja nopeuttaa algoritmin toimintaa. Peruseriaate on kummassakin aliohjelmassa täsmälleen sama ja ne eroavat toisistaan saman tekijän suhteen kuin edellisessä kappaleessa esitetyn algoritmin kaksi eri versiota. Jos siis tulostettava materiaali muodostuu useista samanlaisista tiedostoista, tarvitsee niiden koko selvittää ainoastaan kerran, muussa tapauksessa kyseinen muuttuja on selvitettävä erikseen jokaisen tiedoston kohdalla.

Kun tiedoston koko tunnetaan, seuraava tarvittava tieto on käytetyn työaseman suorituskyky eli käytännössä tiedostonsiirtonopeus palvelimelle. Kunkin lähetettävän tiedoston kohdalla selvitetään tämä siirtonopeus erikseen, koska käsiteltävässä arkkitehtuurissa käytetään useita työasemia. Tiedoston koon ja siirtonopeuden perusteella lasketaan tiedoston siirtämiseen kuluva aika. Tämä aika lasketaan yhteen lähetettävän tiedoston arvioidun tulostusajan kanssa ja näin saadaan lasketuksi kokonaistulostusaika kullekin tiedostolle. Näin laskettuja tiedostojen tulostusaikoja verrataan kutakin vuorollaan kulloinkin pisimmäksi määritettyyn tulostusaikaan. Jos uusin aika on pidempi kuin aiempi pisin aika, muutetaan koko järjestelmän suorituskykyä kuvaavaksi tulostusajaksi tämä viimeisin laskettu aika. Kyseistä operaatiota toistetaan, kunnes kaikki tulostettavat tiedostot on käyty läpi ja kokonaistulostusaika on näin saatu selvitettyä.

## 8.4.3 Kehitettyjen algoritmien testaus

Taulukossa 14 on esitetty useamman työaseman ja useamman tulostimen tulostusympäristössä suoritettujen koetulostusten mitatut ja edellä kuvatuilla malleilla ennustetut tulostusajat. Näiden erotuksesta on laskettu virheprosentti kaavalla 3. Testisarjat on numeroitu juoksevasti ja kunkin testisarjan sisältämät tiedostot, niiden koko sekä lukumäärä on tarkemmin kuvattu liitteen 5 taulukossa 5. Kyseisestä taulukosta käy lisäksi ilmi kunkin testisarjan sisältämien tiedostojen lähettämiseen käytetty työasema, johon viitataan juoksevilla numerolla, sekä tulostin, jota kulloinkin on käytetty tulostamiseen. Jokaisen tiedoston kohdalle on merkitty aika, joka kului tiedoston siirtämiseen työasemalta palvelimelle. Näiden aikojen perusteella on laskettu kokonaissuorituskyvyn laskemiseen tarvittava työasema-



kohtainen keskimääräinen siirtonopeus työasemalta palvelimelle. Lopuksi taulukkoon on vielä merkitty kunkin testisarjan lopullinen kokonaistulostusaika, joka on esitetty myös alla olevassa vertailutaulukossa.

**Taulukko 14.** Useamman työaseman ja useamman tulostimen ympäristössä suoritettujen testitulostusten mitatut ja edellä kuvatuilla malleilla ennustetut tulostusajat.

Testisarjan no.	Todellinen tulostusaika [s]	Todellinen tulostusaika [min:s]	Ennustettu tulostusaika [s]	Ennustettu tulostusaika [min:s]	Virhe [%]
I	1362	22:42	1358	22:38	0,29 %
II	347	05:47	342	05:42	1,44 %
III	977	16:17	989	16:29	1,23 %
IV	1363	22:43	1368	22:48	0,37 %
V	1232	20:32	1232	20:32	0,00 %
VI	1306	21:46	1306	21:46	0,00 %
VII	159	02:39	160	02:40	0,63 %
VIII	308	05:08	306	05:06	0,65 %
IX	403	06:43	403	06:43	0,00 %
X	1227	20:27	1232	20:32	0,41 %
Keskimääräinen virhe					0,50 %
Virheen otoskeskihajonta					0,50 %
Virheen otosvarianssi					0,00 %

Kuten taulukosta selvästi nähdään myös tässä tapauksessa mallit ovat erittäin tarkkoja, niiden tekemät virheet ovat keskimäärin 0,5 % luokkaa eli näin pitkillä tulostusajoilla käytännössä merkityksettömiä. Myös virheistä laskettu otoskeskihajonta ja otosvarianssi ovat myös erittäin pieniä eli tuloksia voidaan pitää erittäin luotettavina. Jälleen on muistettava, että näin tarkkojen ennusteiden edellytyksenä on, että järjestelmään kuuluvan yksittäisen tulostimen suorituskyky on riittävällä tarkkuudella ennustettavissa. Taulukossa kuvatut ennusteet ovat siis näin tarkkoja ainoastaan siitä syystä, että mallien testauksessa käytettiin yksittäisten tulostimien suorituskykyennusteiden sijasta tulostusvaiheessa mitattuja aikoja. Tämä ei kuitenkaan muuta sitä tosiasiaa, että edellä kuvatut tulostusjärjestelmän suorituskykyä kuvaavat mallit ovat erittäin tarkkoja ja luotettavia. Niiden toimivuuden edellytyksenä on, että yksittäisten tulostimien suorituskyvyn ennustamiseen kehitettyjen neuroverkkomallien tarkkuutta pystytään parantamaan.

Johtuen käytetystä koeympäristöstä, sisälsi testitulostusjärjestelmä laajimmillaankin ainoastaan kolme työasemaa ja kolme tulostinta. Nämä eivät samanaikaisesti käytettynäkään vielä kuormittaneet verkkoa niin paljoa, että kuormitus olisi merkittävästi vaikuttanut datasiirron tehokkuuteen. Oletettavaa kuitenkin on, että samanaikaisesti käytettävien työasemien ja tulostimien lukumäärän kasvaessa, myös tästä johtuva verkkokuorman kasvaminen vaikuttaa yhä merkittävämmiin muodostuviin kokonaistulostusaikoihin. Tästä syystä verkkokuorman huomiotta jättäminen on selvä puute mallissa, jos sillä ryhdytään mallintamaan laajempia järjestelmiä. Verkkokuormitusta saattaa lisätä myös muu, itse tulostusjärjestelmästä riippumaton verkkoliikenne.

## 9

# MALLIEN SOVELTAMINEN KÄYTÄNNÖSSÄ

Tässä luvussa käydään läpi kehitettyjen mallien soveltamismahdollisuudet niiden tämän hetkisessä muodossa. Lisäksi on pyritty selvittämään, mitä vaatimuksia kehitetyt mallit asettavat tulostusjärjestelmälle siinä vaiheessa, jos niitä ryhdytään soveltamaan laajamittaisesti, esimerkiksi osana tuotannonohjausjärjestelmää. Perusedellytyksenä laajamittaiselle soveltamiselle voidaan pitää, että mallit integroidaan kiinteäksi osaksi tuotannonohjausjärjestelmää eli tämän kaltaisesta sovelluksesta ollaan vielä aika kaukana. Viimeisessä kappaleessa on listattu muutamia ongelmia, joihin hyvin todennäköisesti törmätään siinä vaiheessa kun malleja ryhdytään soveltamaan nykyisessä muodossa tai osana laajempaa kokonaisuutta. Samassa yhteydessä on myös pyritty hieman selvittämään kyseisten ongelmien taustoja ja mahdollisia keinoja ratkaista ne.

## 9.1 Mallien hyödyntäminen nykyisessä muodossaan

Ennenkuin kehitettyjä järjestelmämalleja voidaan ryhtyä hyödyntämään, tulee mallintaa tulostusjärjestelmään kuuluvat yksittäiset tulostimet ja selvittää käytettävien työasemien suorituskyyky. Tulostimien mallintaminen tapahtuu kehitettyjen neuroverkkomallien avulla. Neuroverkkomallien opetusaineisto voi olla keinotekoisesti generoitua kuten tässä diplomityössä, mutta käytettäessä reaalista aineistoa ovat opetetut mallit aina huomattavasti luotettavampia. Tässä vaiheessa on tärkeää tuntea tulostettavan opetusaineiston rakenne ja sisältö mahdollisimman tarkasti. Mitä yksityiskohtaisemmin opetusaineisto tunnetaan ja mitä enemmän sitä on, sitä luotettavampia mallit ovat opetuksen jälkeen. Kannattaa kuitenkin muistaa, että opetusmateriaalia kuvaavat muuttujat on pystyttävä selvittämään myös niistä tiedostoista, joiden tulostusaikoja mallilla halutaan ennustaa eli liian monen syötetiedon käyttäminen malleja opettaessa saattaa soveltamisvaiheessa johtaa ongelmiin. Neuroverkkojen opetukseen kuluu yleensä runsaasti aikaa, mutta niiden hyödyntäminen on sen jälkeen erittäin nopeaa eli varsinaisessa sovellusvaiheessa syötevektoreiden muuntaminen vastevektoreiksi kestää sekunnin murto-osia. Tämä johtuu siitä, että tässä vaiheessa suoritetaan tarvittavat laskutoimitukset ainoastaan kerran, kun ne opetusvaiheessa saatetaan joutua suorittamaan jopa kymmeniätuhansia kertoja.

Yksittäisen tulostimen suorituskyykymallia rakennettaessa tulostettava opetusaineisto on aina lähetettävä samalta työasemalta, koska neuroverkkomallit eivät tarjoa mahdollisuutta ottaa työaseman suorituskyyky huomioon tässä vaiheessa eli se huomioidaan vasta lopullisessa koko järjestelmän mallinnuksessa. Työasemien suorituskyyky eli käytännössä tiedon siirtonopeus työasemalta palvelimen tulostusjonoon voidaan määrittää tulostimien mallintamisen yhteydessä, mutta se on myös helppo määrittää erillisenä työvaiheena. Tiedostoja lähetetään mallinnettavalta työasemalta palvelimelle, mitataan siirtoon kulunut aika ja jaetaan sillä siirretyn tiedoston koko. Siirtoaikaa mitattaessa kannattaa käyttää useita erikokoisia tiedostoja. Näin määritetyistä useista siirtonopeuksista lasketaan keski-



arvo, jolloin saatua tulosta voidaan pitää luotettavampana. Määritetty siirtonopeus on aina työasemakohtainen ja siitä syystä se on yksi hajautettujen järjestelmien suorituskyvyn ennustamiseen käytettyjen algoritmien syötteistä.

Kun nyt tunnetaan käytettävien työasemien tiedonsiirtonopeudet ja käytössä on tulostin-kohtaiset, tulostusajan ennustamiseen soveltuvat neuroverkkomallit, voidaan halutun tulostusarkkitehtuurin suorituskykyä ryhtyä ennustamaan. Ensiksi on selvitettävä tulostettavista Postscript-tiedostoista tulostinkohtaisten neuroverkkojen vaatimat syötetiedot<sup>1</sup>. Nämä syötetään kunkin tulostettavan tiedoston osalta tulostamiseen käytetyn tulostimen neuroverkkomallille. Vasteena saadaan aika, joka oletettavasti kuluisi tiedoston tulostamiseen valitulla tulostimella ja opetukseen käytetyllä työasemalla. Jotta tätä tulostusaikaa voitaisiin hyödyntää mallinnuksessa yleisemmin, tulee siitä pystyä erottamaan opetuksessa käytetyn työaseman ja palvelimen väliseen datasiirtoon kulunut aika.

Tietyn työaseman ja palvelimen välistä tiedonsiirtonopeutta voidaan teoriassa pitää vakiolisenä ja tästä syystä pystytään käytetyn neuroverkkomallin avulla lasketusta ennusteesta helposti erottamaan oletettuun siirtoon kulunut aika. Näin ollen on tulostusaikaennusteita laskettaessa aina tunnettava myös neuroverkkomallin opetuksessa käytetyn työaseman tiedonsiirtonopeus. Tämä ei sinänsä ole ongelma, koska opetusaineiston tulostamisessa voidaan käyttää aina samaa tai sitä ominaisuuksiltaan vastaavaa järjestelmään kuuluvaa työasemaa, jonka ominaisuudet ovat yleisesti tunnettuja. Tämän siirtoajan vähentämisen jälkeen käytössä on työasemariippumaton, tulostimen suorituskykyä kuvaava tulostusaikaennuste, jota voidaan hyödyntää mallinnuksessa riippumatta siitä, mitä työasemaa käytetään.

Seuraava vaihe on käytetystä järjestelmäarkkitehtuurista riippuen sopivan hajautetun järjestelmän mallin valitseminen. Kokonaisen tulostusjärjestelmän suorituskyvyn ennustamiseen kehitettyjen algoritmien pohjalta laadittu ohjelma kysyy käyttäjältä tulostettavien tiedostojen lukumäärän, niiden koot ja arvioidut tulostusajat valituilla tulostimilla, sekä kunkin tiedoston lähettämiseen käytetyn työaseman. Tämän jälkeen ohjelma laskee valitun algoritmin määräämällä tavalla edellä mainittujen syötteiden perusteella järjestelmän kokonaistulostusaikaennusteen. Hajautettujen järjestelmien malleja ei siis ole liitetty osaksi tulostusjärjestelmää ja siitä syystä käyttäjä joutuu syöttämään lähes kaikki laskennassa tarvittavat suuret manuaalisesti ohjelmaan. Ainoastaan järjestelmään kuuluvien työasemien ja tulostimien ominaisuudet on valmiiksi liitetty osaksi ohjelmaa, mutta käyttäjän on itse valittava kulloinkin käytettävä työasema ja tulostin.

## 9.2 Laajamittaisen soveltamisen edellytykset

Aiemmin jo todettiin, että jos kehitettyjä neuroverkkomalleja ja algoritmeja halutaan soveltaa laajemmassa mittakaavassa eli käytännössä tuotantoympäristössä, tulisi ne integroida kiinteäksi osaksi tuotannonohjaus- ja valvontajärjestelmää. Tämä johtuu siitä, että nopeatempoisessa elektronisessa painotuotannossa kehitettyjen mallien hyödyntäminen nykyisessä muodossa on aivan liian hidasta. Tämä johtuu siitä, että suurin osa laskennan vaatimista parametreista joudutaan selvittämään ja käsittelemään manuaalisesti erillisinä työvaiheina. Lopputuloksena saatetaan ajautua tilanteeseen, jossa tulostusaikaennusteen laskemiseen kuluu enemmän aikaa kuin itse tulostamiseen. Ideaalisessa tilanteessa tulostusjärjestelmään syötetään ainoastaan tulostettava Postscript-tiedosto ja haluttu kappalemäärä. Järjestelmä selvittää tämän jälkeen automaattisesti tarvittavat parametrit ja tulostusjärjestelmän sen hetkisen kuormituksen. Näiden pohjalta se suorittaa tarvittavat laskutoimitukset ja lopuksi antaa palautteena arvion siitä, kuinka kauan työn valmistumi-

---

1. Jos käytetään yleistä tulostinmallia, joudutaan syöttämään myös tulostinta kuvaavat parametrit.



seen kuluu aikaa. Seuraavassa on lyhyesti pyritty selvittämään, mitä vaatimuksia tämän tyyppiselle järjestelmälle asetetaan.

Kehitetyt hajautettujen järjestelmien mallit edellyttävät, että yksittäisten tulostimien suorituskky on mallinnettu ennen kuin koko järjestelmän suorituskkyä ryhdytään ennustamaan. Yksittäisten tulostimien suorituskkyyn ennustamisen käytetään neuroverkkomalleja ja jotta näitä voitaisiin opettaa tehokkaasti, tulisi tulostusjärjestelmän kautta pystyä keräämään opetukseen tarvittavaa informaatiota ja suorittamaan varsinaisen opettamisen automaattisesti. Malleja tulisi siis pystyä opettamaan sitä mukaan, kun lisää opetusmateriaalia on saatavilla. Suurin osa neurolaskentaa tällä hetkellä hyödyntävistä sovelluksista on toteutettu juuri näin eli niitä opetetaan koko ajan niiden ennustustarkkuuden parantamiseksi. Opetuksen tulisikin tapahtua siten, että se ei häiritse varsinaista mallien hyödyntämistä. Tämä ei sinänsä ole ongelma, koska opetuksen ei tarvitse olla reaaliaikaista, eli se voisi tapahtua esimerkiksi yöaikana kun järjestelmän kuormitus on alhainen.

Neuroverkkojen opetukseen tarvittavat parametrit olisi pystyttävä selvittämään tulostetuista tiedostoista automaattisesti, jotta järjestelmä olisi toimiva. Lisäksi järjestelmän kautta on kyettävä rekisteröimään kunkin tiedoston lopullinen tulostusaika. Hajautetun painatusjärjestelmän suorituskkyyn kannalta on myös olennaisen tärkeää tuntea sekä opetuksessa, että varsinaisessa tulostuksessa käytettyjen työasemien ja tulostinpalvelimen välinen datasiirtonopeus, jotta ennusteet pystytään laskemaan tarkasti. Datasiirtonopeuden selvittäminen ei sinänsä ole vaikeaa, koska se on aina työasemakohtaisesti vakio ja yksinkertaisten testitulostusten avulla laskettavissa. Se tarvitsee selvittää jokaiselle työasemalle ainoastaan kerran jos järjestelmä pysyy muuttumattomana, eli uudelleen määrittäminen on tarpeen ainoastaan siinä tapauksessa, että tulostusjärjestelmässä tehdään muutoksia.

Kaikki edellä kuvattu on valmistelevaa toimintaa ja varsinaisesti mallien hyödyntämisen voidaan katsoa alkavan sillä hetkellä, kun tulostettava Postscript-tiedosto syötetään tulostusjärjestelmään. Ensimmäinen tehtävä on selvittää tiedostosta neuroverkkomallien vaatimat syöteparametrit. Opetusvaiheen vaatima tiedostojen analysointi ei välttämättä ole aikakriittistä, koska se voidaan toteuttaa off-line toimintona, mutta varsinaisessa tuotannossa tiedoston analysoinnin on tapahduttava mahdollisimman nopeasti, eli analysointivaihe ei saa merkittävästi hidastaa varsinaista tulostusprosessia. Tämän tyyppiseen analysointiin tarvitaan ohjelmisto, joka kykenee nopeasti selvittämään tulostettavan Postscript-tiedoston sisältöä ja rakennetta kuvaavat parametrit. Kun parametrit on selvitetty, järjestelmän tulee pystyä selvittämään tiedoston tulostamiseen soveltuvien tulostimien tila eli onko tulostin vapaana ja voidaanko sille lähettää tiedostoja. Myös työn lähettämiseen käytetyn työaseman suorituskkyyn tulisi olla automaattisesti selvitettävissä tai järjestelmään valmiiksi syötetty.

Kun kaikki edellä mainitut parametrit on selvitetty, voidaan laskea neuroverkkomalleilla tulostinkohtaiset tulostusaikaennusteet niille tulostimille, jotka järjestelmässä soveltuvat tulostukseen ja joita on tarkoitus hyödyntää. Näiden ennusteiden pohjalta voidaan kehitettyjen järjestelmäalgoritmien avulla laskea järjestelmän suorituskkyä kuvaava kokonaistulostusaikaennuste. Tässä vaiheessa voidaan useampia tulostimia hyödynnettäessä ottaa huomioon myös tulostimien mahdolliset suorituskkyerot. Nopeammille tulostimille lähetetään useampia tiedostoja kuin hitaammille ja näin voidaan optimoida tulostusta siten, että kaikilla käytetyillä tulostimilla työt valmistuvat mahdollisimman samanaikaisesti. Samassa yhteydessä huomioidaan ennusteessa myös mahdolliset muut työt, joiden käsittely saattaa viivästyttää uusimman työn käsittelyä. Tämä on sinänsä helppo toteuttaa, jos järjestelmä on täysin integroitu. Kaikille järjestelmään jo syötetyille töille on vastaavalla tavalla laskettu tulostusaikaennuste eli järjestelmässä on ainoastaan pidettävä kirjaa käsiteltävistä ja valmistuneista tulostustöistä.



## 9.3 Ongelmat

### 9.3.1 Syöteparametrien selvittäminen tiedostoista

Neuroverkkomallien opetuksessa ja tulostusaikojen ennustamisessa tarvitaan runsaasti eksaktia informaatiota tulostetun/tulostettavan tiedoston sisällöstä ja rakenteesta. Yleensä Postscript tiedostosta on kuitenkin ilman suurempia toimenpiteitä selvitettävissä ainoastaan tiedoston koko ja joskus sivumäärä. Muiden tarvittavien tietojen selvittäminen on erittäin vaikeaa nykyisin käytössä olevilla menetelmillä. Tehtävää hankaloittaa erityisesti, että kaikki sovellukset generoivat sivunkuvauksen omilla ajureillaan eli kaksi sovellusta saattaa luoda samasta sivuinformaatiosta kaksi täysin toisistaan poikkeavaa sivunkuvausta. Tällöin tietyille dokumentin elementeille ominaisten piirteiden erottaminen koodista voi olla erittäin hankalaa, koska niiden esitysmuoto ei ole sovellusriippumatonta. Näin ollen Postscript-koodia analysoitaessa tulisi pystyä ottamaan huomioon sen generointiin käytetyn sovelluksen ominaisuuksia. Tämä taas johtaa helposti siihen, että analysointivaiheessa tiedosto joudutaan ikään kuin rasteroimaan ylimääräisen kerran, jotta analyysistä tulisi riittävän tarkka. Rasterointi on kuitenkin eniten aikaa kuluttava työvaihe tulostusprosessissa ja näin ollen aivan liian työläs ja hidas operaatio suoritettavaksi kahteen kertaan. Yksi ratkaisu saattaisi olla sovelluskohtaisten analysointityökalujen kehittäminen, mikä olisi kuitenkin suhteellisen mittava ja vaativa urakka.

### 9.3.2 Verkkokuorman huomioiminen

Kuten aiemmin on useaan kertaan todettu, edellä kuvatut hajautettujen painatusjärjestelmien mallit pätevät ainoastaan siinä tapauksessa, että verkon kuormitus ei kasva niin suureksi, että se vaikuttaa teoreettisesti laskettuihin suorituskykyarvoihin. Jos liikennettä on verkossa niin paljon, että siirtonopeudet putoavat tai verkko menee kokonaan tukkoon, eivät mallit enää pidä paikkaansa ja ennusteiden tarkkuus kärsii. Verkkokuorman huomioiminen voi olla mahdollista vielä siinä vaiheessa, kun toimitaan lähiverkkoympäristössä ja kaikki verkkokuorma aiheutuu itse tulostusjärjestelmän sisäisestä datasiirrosta. Tämän tyyppistä datasiirtoa on työaseman ja palvelimen, sekä palvelimen ja tulostimen välinen tiedonsiirto. Tilanne monimutkaistuu välittömästi kun verkossa on jonkin tulostusjärjestelmän ulkopuolisen tekijän aiheuttamaa kuormitusta tai verkkoympäristö on lähiverkkoa laajempi, esimerkiksi MAN-verkko. MAN-verkkojen tapauksessa datasiirron tehokkuuteen vaikuttavat kuormituksen lisäksi esimerkiksi yhteydenmuodostusmekanismit ja reitityksen toteutustapa.

Tulostusjärjestelmään kuulumattomien verkkolaitteiden aiheuttamaa kuormitusta on lähes mahdoton ennakoida, koska eritoten lähiverkoissa hyvin harvoin muodostuu selviä ennustettavia kuormituspiikkejä. Tämä johtuu siitä, että kaikki lähiverkon käyttäjät hyödyntävät verkkoa hyvin sattumanvaraisesti. Huomattavasti laajemmissa ympäristöissä voidaan esimerkiksi vuorokausisyklistä erottaa selviä huippukuormitusjaksoja, mutta niidenkin huomioiminen tulostusaikaennusteissa on erittäin vaikeaa, koska ne lasketaan yleensä keskimääräisenä kuormituksena aikayksikköä kohden eli niidenkään välittämä informaatio ei ole riittävän eksaktia matemaattisessa mallissa hyödynnettäväksi. Positiivista tilanteesta on, että siinä vaiheessa kun todellisia kaupallisia tietokoneverkkojen kautta hajautettuja painatusjärjestelmäympäristöjä ryhdytään kehittämään, hyödynnettävät siirtomediat ovat kapasiteetiltaan täysin eri luokkaa kuin tämän diplomityön testiympäristössä. Tällöin tiedonsiirtonopeus muodostuu järjestelmän kokonaissuorituskyvyn kannalta marginaaliseksi eli mahdollinen ennustamaton viive tiedonsiirrosta ei enää merkittävästi vääristä laskettua ennustetta.



## 10

## MAHDOLLINEN JATKOKEHITYS

Tärkeintä jatkokehityksen kannalta olisi päästä testaamaan kehitettyjä malleja reaaliseen tuotantoympäristöön ja sitä kautta mahdollisesti integroimaan ne osaksi todellista tulostusjärjestelmää. Tämä edellyttää runsaasti ohjelmointityötä eli kehitetyt neuroverkkomallit ja algoritmit tulisi koodata siten, että ne pystyvät toimimaan mahdollisimman automaattisesti kiinteänä osana tulostusjärjestelmää. Järkevintä olisi pyrkiä yhteistyöhön jonkin tuotannonohjausjärjestelmiä tuottavan ohjelmistotalon kanssa. Pelkkä mallien integrointi ei kuitenkaan välttämättä vielä riitä, koska Matlab on laskentateholtan aivan liian hidas sovellus käytettäväksi valmiin sovelluksen osana. Matlab-ohjelman avulla muodostetut neuroverkkomallit tulisi koodata tehokkaampaan muotoon jollakin ohjelmointikielellä. Lisäksi on huomattava, että mallit on kehitetty hyvin suppeassa ympäristössä ja niiden soveltuvuutta huomattavasti laajempien järjestelmien mallintamiseen ei ollut mahdollista tämän diplomityön puitteissa tutkia. Tästä syystä on luultavaa, että jos malleja halutaan hyödyntää laajemmin, tulee niissä ottaa huomioon huomattavasti yleisemmin erilaisia sovellus- ja järjestelmäkohtaisia ominaisuuksia. Seuraavassa on lyhyesti esitetty joitakin esiin tulleita asioita, jotka luultavimmin vaativat lisäselvitystä.

Kaikki opetus- ja testimateriaalina käytetyt Postscript-tiedostot generoitiin käyttäen samaa sovellusta eli Frame Maker 4.02:ta. Sävykuvatiedostojen käsittely suoritettiin Adobe Photoshop 2.5.1:llä ja viivakuvia puolestaan käsiteltiin ainoastaan Corel Draw 4.0:lla. Tästä on seurauksena, että varmuudella ei voida sanoa, mikä on käytettävän sovelluksen vaikutus Postscript-koodin ominaisuuksiin ja tehokkuuteen. Aiempien kokemusten perusteella voidaan olettaa, että joissakin tapauksissa voi esiintyä selviä eroja samasta materiaalista eri sovelluksilla generoitujen Postscript-tiedostojen tulostusaikojen välillä. Mielenkiintoista olisi selvittää, kuinka suuria erot tulostusajoissa kahdella eri sovelluksella samasta lähdemateriaalista generoitujen Postscript-tiedostojen välillä voivat olla. Lisäksi jos erot havaittaisiin merkittäviksi, miten ne olisi mahdollista huomioida mallinnuksessa, vai olisiko jokaista sovellusta varten opetettava oma neuroverkkomallinsa.

Neuroverkkomalleissa syötteinä käytettyjen materiaaliominaisuuksien lisäksi tulostettavat tiedostot sisältävät runsaasti muita elementtejä, joilla voidaan olettaa olevan ainakin jonkin asteista vaikutusta muodostuviin tulostusajoihin. Tällaisia elementtejä voivat olla esimerkiksi kaavat, taulukot ja toistuvasti sivuilla käytetty grafiikka<sup>1</sup>. Myöskään palstojen lukumäärän vaikutusta tulostusajoihin ei tässä työssä tutkittu lainkaan. Se saattaa kuitenkin olla yksi suurimmista syistä neuroverkkomallien testauksessa käytetyn *opetus\_ops*-tiedoston selvästi ennusteista poikkeaville tulostusajoille. Vaikka edellä mainituilla, tässä työssä huomiotta jätetyillä elementeillä ei valmiita Postscript-tiedostoja käsiteltäessä olisi-kaan vaikutusta tulostusajoihin, viimeistään siinä vaiheessa kun Postscript käännökseen kuluva aika huomioidaan järjestelmän tehokkuutta määriteltäessä ne muuttuvat erittäin merkittäviksi. Onhan selvää, että mitä monimutkaisempaa sivunkuvausta muodostetaan, sitä kauemmin siihen kuluu aikaa.

Itse asiassa kaikki edellä mainitut mahdollisesti tulostusajoihin vaikuttavat tekijät voidaan koota yhden yhteisen nimittäjän alle, joka on Postscript. Jotta Postscript-sivunkuvaus-

1. Esimerkiksi rajauksessa käytetyt viivat, kehykset ja niin edelleen.



kieltä hyödyntävien tulostusjärjestelmien suorituskykyä voitaisiin ennustaa tehokkaasti, tulisi pystyä automaattisesti analysoimaan tulostettavien Postscript-tiedostojen rakenne ja ominaisuudet. Tämä ei kuitenkaan vielä tällä hetkellä ole mahdollista ja tätä voidaan pitää yhtenä suurimmista esteistä tässä diplomityössä kuvatun kaltaisten mallien hyödyntämiselle käytännössä. Merkittävä edistysaskel olisi, jos pystyttäisiin rakentamaan järjestelmä, joka sovellus- ja järjestelmäriippumattomasti pystyisi selvittämään Postscript-tiedoston sisältämät elementit ja niiden datamäärät. Näiden parametrien selvittäminen on mallien toiminnan kannalta ensiarvoisen tärkeää.

Tässä diplomityössä käsiteltiin ainoastaan Postscript-kieltä, koska se on kaikkein yleisin nykyään käytetyistä tulostimien ohjaukseen soveltuvista sivunkuvauskielistä ja lisäksi se on hyvin pitkälle laitteistoriippumatonta. Käytössä on myös monia muita nimenomaan tulostimien ohjaukseen suunniteltuja sivunkuvauskieliä. Tällaisia ovat esimerkiksi Hewlett Packardin PCL/HPGL ja Xeroxin Interpress. Nämä ovat yleensä sidottuja tietyn valmistajan tulostimiin, mutta esimerkiksi PCL on erittäin yleisesti käytössä toimistoympäristöjen ulkopuolella. Tähän vaikuttaa hyvin pitkälti tulostimien hinta eli Postscript-tulostimet ovat huomattavasti kalliimpia kuin pelkkää PCL:ää käsittelevät tulostimet. Myös Interpressillä on tällä hetkellä erittäin merkittävä asema, koska sitä käytetään Xeroxin elektronisen painokoneen Docutechin sivunkuvauskielenä. Tästä syystä olisi ehkä paikallaan tutkia myös näiden sivunkuvauskielten ominaisuuksia ja vaikutusta muodostuviin tulostusaikoihin. Jos tässä työssä kehitetyt mallit osoittautuvat epätarkeiksi, tulisi ehkä rakentaa suorituskyky-malleja myös näitä sivunkuvauskieliä hyödyntäville tulostimille ja tulostusjärjestelmille.

Merkittäväksi sovelluskohteeksi voisi osoittautua myös järjestelmä, jossa suurikokoinen Postscript-tiedosto jaettaisiin pienempiin palasiin ja jokainen palanen käsiteltäisiin prosessissa erillisenä kokonaisuutena. Näin ollen yhden suuren tiedoston käsittely voitaisiin jakaa useammille tulostimille/rasterointiprosessoreille ja näin hyödynnettäisiin rinnakkaislaskentaa. Tässä diplomityössä tulostettiin ainoastaan kokonaan erillisiä tiedostoja useille tulostimille samanaikaisesti, ei jaettuja tiedostoja. Kuvatun kaltaisen järjestelmän toteuttaminen vaatisi kuitenkin sellaisen sovelluksen kehittämistä, joka pystyisi jakamaan suuren Postscript-tiedoston osiin siten, että näiden käsittely erillisinä olisi mahdollista. Vielä ei ole onnistuttu kehittämään järjestelmiä jotka kykenisivät tähän. Ongelmaa yritetään kuumaisesti ratkaista useissa tutkimus- ja tuotekehitysprojekteissa, joten tämän tyyppisen jatkokehityksen voidaan katsoa tietyssä mielessä olevan resurssien tuhlaamista.

Selvä jatkotutkimuksen kohde on myös itse materiaalin kuvaamiseen käytetyt parametrit ja itse kuvaamistapa. Tässä työssä on tutkittu ainoastaan yhtä tapaa kuvata tulostettavan materiaalin sisältöä neuroverkoille, eli on laskettu erityyppisten elementtien osuuksia kokonaisdatamäärästä. Siihen, kuvaavatko valitut elementit tulostettavaa materiaalia parhaalla mahdollisella tavalla ei varsinaisesti otettu kantaa. Diplomityössä ei myöskään ollut mahdollisuutta perehtyä Postscript-kielen rakenteeseen ja ominaisuuksiin siinä määrin kuin se olisi ollut tarpeellista. Tulisikin tutkia, voidaanko Postscript-koodista suoraan selvittää neuroverkkojen syötteiksi soveltuvia tunnuslukuja, joiden avulla olisi mahdollista ennustaa tiedoston tulostusaikaa. Tämä helpottaisi mallien soveltamista huomattavasti, koska elementtien datamääräosuuksien selvittäminen Postscript-tiedostosta ilman alkupe-  
räistä aineistoa on erittäin hankalaa.

Verkkokuorman huomioiminen malleissa on myös erittäin ongelmallista, koska sen ennustaminen on vaikeaa. Ratkaisuksi voisi osoittautua esimerkiksi erilaisten jo kehitettyjen todennäköisyyksiin ja tilastomatematiikkaan perustuvien tietokoneverkkomallien integroiminen osaksi malleja. Myös jonoteorioiden hyödyntäminen saattaisi tulla kysymykseen. Nämä eivät kuitenkaan tarjoa mahdollisuutta tarkkoihin ennusteisiin, mutta jos verkko-  
kuormituksen havaitaan vääristävän tuloksia, kyseisten mallien avulla niitä voidaan ainakin tarkentaa huomattavasti. Parhaiten todennäköisyyteen ja tilastoihin perustuvat mallit toimivat ympäristöissä, joissa mahdollisten samanaikaisten käyttäjien määrä on rajallinen (esimerkiksi Token ring ja FDDI). Ethernet-protokollan tapauksessa ennustaminen on taas hankalampaa, koska kaikilla on samanaikaisesti mahdollisuus hyödyntää samaa mediaa.



## 11

## JOHTOPÄÄTÖKSET

Tällä hetkellä näyttää hyvin todennäköiseltä, että 1-värisen elektronisen painatuksen osuus graafisen teollisuuden tuotannosta tulee muutaman seuraavan vuoden aikana edelleen kasvamaan merkittävästi. Tulostinvalmistajat tuovat markkinoille yhä nopeampia tulostimia, jotka alkavat myös tulostusjäljen laadun puolesta olla riittävän korkeatasoisia useimmille painotuotteille. Myös tulostimien hinta ja käyttökustannukset ovat alentuneet siinä määrin, että ne pystyvät kilpailemaan konventionaalisen painatuksen kanssa. Lisäksi myös kilpailukykyisiä elektronisia 4-väripainokoneita alkaa pikku hiljaa olla markkinoilla, joten myös sillä alueella tulee kilpailu lähivuosina lisääntymään. Tällä hetkellä arviot kannattavasta painoskoosta suurtehotulostimilla liikkuvat 1-väripainatuksessa 1000...3000 ja 4-väripainatuksessa 800...1500 kappaleen välillä. Oletettavaa kuitenkin on, että jo lähitulevaisuudessa teknisen kehityksen myötä elektronisilla painokoneilla tullaan painamaan yhä suurempia painoksia.

Koska suurteholuokan elektroniset painokoneet ovat edelleen miljoonainvestointeja on erittäin todennäköistä, että elektronisia painatusjärjestelmiä tullaan toteuttamaan yhä useammin hajautetusti. Tämän tyyppinen järjestelmä muodostuu useista alemman teholuokan tulostimista, jotka on yhdistetty datasiirtoverkon kautta yhdeksi kokonaisuudeksi. Näin käytössä olevaa kapasiteettia saadaan kasvatettua ja lisäksi varsinainen tulostusoperaatio tapahtuu mahdollisesti lähempänä varsinaista loppukäyttäjää; näin säästetään esimerkiksi kuljetuskustannuksissa. Lopullisena tavoitteena voidaan pitää järjestelmiä, joissa painatus tapahtuu suoraan loppukäyttäjän henkilökohtaisella tulostimella. Tämän tyyppisten järjestelmien hallitseminen on hankalaa muun muassa siitä syystä, että niiden kapasiteetin ja suorituskyvyn arvioiminen on vaikeaa. Järjestelmään saattaa kuulua useita erityyppisiä tulostimia ja käytettyjen tiedonsiirtoyhteyksien suorituskyky voi myös vaihdella.

Aiemmissa tutkimuksissa on jo käynyt ilmi, että tämän tyyppisen elektronisen painatusjärjestelmän mallintaminen perinteisillä menetelmillä ei onnistu. Suurimpana syynä tähän on pidetty järjestelmään kuuluvien komponenttien sekä tulostettavan materiaalin sisältämien elementtien välisten riippuvuuksien epälineaarisuutta. Tässä työssä havaittiin, että suurin epälineaarisuuden lähde on nimenomaan tulostettavan materiaalin sisältämät elementit, joiden yhteisvaikutukset muodostuviin tulostusaikoihin ovat erittäin vaikeasti ennustettavissa. Erityisesti voidaan mainita EPS-sävykuvia sisältävät tiedostot, joiden todelliset tulostusajat poikkesivat ennusteista erittäin selvästi kaikkien testitulostuksissa käytettyjen tulostimien tapauksessa. Tämän voidaan olettaa johtuvan siitä, että tulostusjärjestelmien pullonkaula on edelleen rasterointiprosessori, jonka toimintaan tulostettavien tiedostojen sisältö vaikuttaa ratkaisevasti. Selvästi oli myös havaittavissa, että sivumäärällä on koneistonopeudeltaan hitaiden tulostimien tapauksessa Postscript-tiedoston koon perusteella laskettuja lineaarisia ennusteita vääristävä vaikutus. Käytännössä tämä tarkoittaa, että tiedostot tulostuvat hitaammin kuin mitä niiden koko antaisi olettaa. Vaikka edellä mainitut olivat selvimmin havaittavissa olevat piirteet, myös muilla tiedostojen sisältämällä elementeillä havaittiin olevan selvä vaikutus muodostuviin tulostusaikoihin.

Koska perinteisten menetelmien avulla ei riittävän tarkkoja suorituskykymalleja ole pystytty kehittämään, päätettiin tässä työssä hyödyntää niin sanottuja oppivia ja älykkäitä menetelmiä, joista päädyttiin käyttämään neurolaskentaa. Tämä on perusteltua siitä



syystä, että se soveltuu erityisesti suuresta datamäärästä vaikeasti havaittavien epälineaaristen riippuvuuksien etsimiseen. Neuroverkkojen toiminta perustuu siihen, että niille opetetaan esimerkkien avulla järjestelmän toimintaa kuvaavat riippuvuudet eli tässä tapauksessa tiedoston koon, sivumäärän ja sen sisältämien elementtien vaikutus muodostuviin tulostusaikoihin. Neuroverkkojen voidaan suoritettujen kokeiden perusteella sanoa soveltuvan tulostimien mallintamiseen, mutta tarkkoihin ennusteisiin pääseminen edellyttää, että käytössä on runsaasti itse tulostustapahtuman tehokkuutta kuvaavaa opetusmateriaalia. Lisäksi tarkkoihin ennusteisiin pääseminen edellyttää tulostettavan materiaalin mahdollisimman tarkkaa kuvaamista eli mitä tarkempi on kuvaus, sitä tarkempi on mallin ennuste. Tätä väittämää tukee testeissä havaittu piirre, että keskimääräinen virhe, sekä virheistä laskettu otoskeskihajonta ja otosvarianssi pienenevät selvästi sitä mukaa, kun materiaalin sisällön kuvausta tarkennetaan.

Tässä työssä kehitettyjen neuroverkkomallien pohjalta lasketut tulostinkohtaiset ennusteet eivät sinänsä olleet kovin luotettavia. Virhelähteet olivat kuitenkin selvästi osoitettavissa, joten malleja edelleen kehittämällä päästään varmasti huomattavasti parempiin tuloksiin. Ongelmien voidaan olettaa johtuvan osaksi siitä, että opetuksessa käytettyä materiaalia oli liian vähän suhteessa riippuvuuksien monimutkaisuuteen. Näin ollen mallit opetuksen jälkeen kykenivät havaitsemaan vain yksinkertaisimmat riippuvuudet tiedostojen ominaisuuksien ja tulostusaikojen välillä. Mallien epätarkkuus saattaa näin ollen osittain johtua siitä, että jo alunperin lähdettiin rakentamaan liian kunnianhimoista ja yleistä järjestelmämallia eli opetusmateriaalista tehtiin liian monimutkaista verrattuna sen määrään. Mutta kuten sanottua, tilanteen tulisi korjaantua kun malleja opetetaan edelleen. Neuroverkkojen onnistuneimmat sovellukset on rakennettu nimenomaan siten, että malleja tarkennetaan koko ajan eli opetusta jatketaan läpi koko sen ajan, kun niitä hyödynnetään.

Neuroverkkomallien testauksessa päädyttiin käyttämään samaa testiaineistoa kuin Pertti Mäkitalo omassa diplomityössään. Tätä voidaan tietystä mielestä pitää huonona ratkaisuna. Vaikka tällä tavoin mahdollistettiin vertailu diplomitoissa kehitettyjen mallien välillä, samalla todennäköisesti vaikutettiin neuroverkkomallien ennustustarkkuuden testausmahdollisuuksiin. Tämä johtuu siitä, että ainoa mahdollisuus testata neuroverkkomallien luotettavuutta on käyttää opetusaineistosta poikkeavaa testiaineistoa, jonka tulisi opetusaineiston tavoin olla mahdollisimman kattavaa ja vaihtelevaa. Nyt käytetty materiaali ei täysin vastaa näitä vaatimuksia, vaikka se edustaa sisällöltään ja rakenteeltaan yleisimmin tulostettuja dokumenttityyppejä. Tässä diplomityössä opetetut neuroverkkomallit ennustavat parhaalla mahdollisella tavalla ainoastaan juuri tämän tyyppisen materiaalin tulostusaikoja. Tilanteen voidaan kuitenkin olettaa korjaantuvan kun mallien testaamiseen käytetään reaalista, todellisessa tuotantoympäristössä tulostettua vaihtelevampaa materiaalia.

Edellä mainittujen tulostinkohtaisesti opetettavien neuroverkkomallien lisäksi muodostettiin kaksi mallia, joiden syötteet ovat hieman näistä poikkeavia. Ensimmäisen tapauksessa syötteinä käytettiin ainoastaan tiedoston kokoa, sivumäärää ja EPS-sävykuvien osuutta kokonaisdatamäärästä. Koska näiden parametrien todettiin selvimmän aiheuttavan poikkeamia lineaarisilla malleilla laskettuihin ennusteisiin, haluttiin selvittää, olisiko pelkäämään näiden parametrien avulla mahdollista riittävällä tarkkuudella ennustaa tulostusaikoja. Tulokset eivät kuitenkaan olleet kovin hyviä ja oletettavaa on, että mitä tarkempi tulostettavan tiedoston sisältökuvaus on, sitä tarkempi on myös neuroverkolla laskettu ennuste. Kyseinen malli jouduttiin edelleenkin laskemaan tulostinkohtaisesti.

Toisessa tapauksessa edellä mainituista malleista tarkimman tuloksen antaneen syötteisiin lisättiin käytettävää tulostinta kuvaavia parametrejä. Parametreiksi valittiin

- koneistonopeus (sivua/minuutissa),
- työmuistin koko (Mt),
- tulostusresoluutio (dpi) ja
- rasterointiprosessorin teho, jota kuvattiin sen kelloaajuudella (MHz).



Muodostetulla mallilla voidaan teoriassa ennustaa minkä tahansa järjestelmään kuuluvan tai siihen liitettävän tulostimen tulostusaikoja, kunhan siitä tunnetaan edellä mainitut parametrit. Opetetun yleisen tulostinmallin ennustustarkkuus oli hyvin lähellä erillisinä opetettujen mallien tulostinkohtaisten mallien tarkkuutta. Näin ollen valitut parametrit kuvaavat tulostimien suorituskykyä riittävän tarkasti. Tästä voidaan päätellä, että on mahdollista melko helposti kehittää yleisiä järjestelmäkohtaisia tulostinmalleja, joiden avulla voidaan ennustaa mielivaltaisen järjestelmään liitettävän tulostimen suorituskykyä esimerkiksi jo ennen sen hankkimista. Tämä helpottaa mallien soveltamista myös siinä mielessä, että kaikkia järjestelmään liitettyjä tulostimia ei välttämättä tarvitse mallintaa erikseen, vaan riittää kun muodostetaan yleinen malli, jonka opetuksessa käytetään riittävän kattavaa leikkausta järjestelmään liitetyistä tulostimista.

Alunperin tarkoituksena oli ennustaa neuroverkkojen avulla myös kokonaisten tulostusjärjestelmien suorituskykyä, mutta tämä lähestymistapa todettiin epäkäytännölliseksi ja ajatuksesta luovuttiin. Jos neuroverkkomalleja sovellettaisiin järjestelmäkohtaisesti, olisi niiden hyödyntäminen yleisemmin erittäin hankalaa ja aikaa vievää: esimerkiksi opetus jouduttaisiin suorittamaan erikseen jokaiselle uudelle tai muuten aiemmista poikkeavalle tulostusarkkitehtuurille. Tästä syystä päädyttiin ratkaisumalliin, jossa mallinnetaan neuronilaskennan avulla ainoastaan kukin järjestelmään kuuluvista tulostimista erikseen. Näitä tulostinkohtaisia malleja hyödyntämällä lasketaan edelleen kehitettyjen suorituskykyalgoritmien avulla tulostusjärjestelmän kokonaissuorituskyky. Tämä on perusteltua siitä syystä, että näin voidaan järjestelmän mallintamisvaiheessa ottaa huomioon tapauskohtaisesti järjestelmän rakenne ja sen sisältämien komponenttien ominaisuudet.

Tulostusarkkitehtuurien mallintamiseen kehitetyt algoritmit osoittautuivat tulostinkohtaisista neuroverkkomalleista poiketen erittäin tarkoin. Ennusteiden ja todellisten tulostusaikojen erotuksesta lasketut virheprosentit olivat keskimäärin alle prosentin luokkaa. Käytännössä tämän suuruinen virhe on merkityksetön eli tulostusjärjestelmät voidaan mallintaa kehitetyillä lineaarisilla algoritmeilla. Lisäksi kehitettyjä algoritmeja voidaan pitää neuroverkkomalleja parempina siitä syystä, että ne eivät aseta mitään rajoituksia järjestelmään liitettyjen työasemien tai tulostimien lukumäärälle. Algoritmeissa erityyppisten laitteiden lukumäärää voidaan käsitellä muuttujana. Yhdellä tulostimella tulostettavien tiedostojen lukumäärällä ei myöskään ole mallien kannalta merkitystä eli myös se on huomioitu malleissa muuttujana. Huomiota kannattaa kiinnittää myös siihen, että vaikka monimutkaisimmassa arkkitehtuurissa tulostettiin kolmelta työasemalta samanaikaisesti kolmelle tulostimelle, ennusteet olivat yhtä tarkkoja kuin yksinkertaisempien arkkitehtuurien malleilla. Tämä viittaa siihen, että ainakaan vielä tämän kokoisessa järjestelmässä ei kasvanut verkkokuorma merkittävästi vaikuttanut tulostusaikoihin.

Ainoa seikka, jonka voidaan olettaa aiheuttavan virhettä hajautetun painatusjärjestelmän lineaaristen mallien avulla laskettuihin ennusteisiin on verkon kuormitus. Selvää on, että mitä enemmän verkossa on kuormaa, sitä hitaampaa on datasiirto työasemalta palvelimelle ja sieltä edelleen tulostimelle. Verkkokuorman ennustaminen erityisesti lähiverkoissa on melko ylivoimainen tehtävä, koska varsinaisia kuormituspiikkejä ei muodostu. Kuorma riippuu hyvin pitkälle käyttäjien satunnaisesti muodostuvista tarpeista hyödyntää verkon tarjoamia palveluja. MAN- ja WAN-ympäristöissä voidaan kuormituksessa mahdollisesti osoittaa tiettyjä huippuja, jotka ovat riippuvaisia lähinnä vuorokauden ajasta. Tämän tyyppisten kuormituspiikkien huomioimista malleissa voidaan pitää hieman arveluttavana, koska niiden ennustaminen perustuu hyvin pitkälle todennäköisyyteen eli kyse ei ole mistään eksaktista parametrista.



## 12

## YHTEENVETO

Muutaman viime vuoden aikana elektroninen painaminen on lisääntynyt räjähdysmäisesti ja kasvun on ennustettu edelleen jatkuvan. Elektroninen painaminen voidaan määritellä painatukseen rinnastettavaksi tulostamiseksi eli tulostimia käytetään ikään kuin painokoneina. Tulostin on tässä yhteydessä hieman hämäävä käsite, koska painokoneina käytetyt pääasiassa elektrofotografiatekniikkaan perustuvat tulostimet ovat suurtehotulostimia, joilla on hyvin vähän yhteistä perinteisten tulostimien kanssa. Ne ovat suorituskyvyltään perinteisiä tulostimia selvästi tehokkaampia ja niihin on usein integroitu erilaisia on-line jälkikäsittely-yksiköitä. Nämä elektroniset painokoneet vaativat konventionaalisten painokoneiden tavoin usein miljoonainvestointeja ja tästä syystä on ryhdytty tutkimaan mahdollisuutta toteuttaa painatusjärjestelmiä hajautetusti. Tällöin suurtehotulostimen sijasta tulostuksessa käytetään useita huomattavasti edullisempia, alemman teholuokan tulostimia eli tulostimien alhaisempi suorituskyyky korvataan niiden lukumäärää lisäämällä.

Hajautettuja painatusjärjestelmiä voidaan muodostaa monella tavalla ja monilla tekniikoilla, mutta tässä työssä termillä hajautettu painatus tarkoitetaan tulostamista, joka tapahtuu useammalle tulostimelle tietokoneverkon kautta. Rajaus katsottiin tarpeelliseksi suorittaa näin, koska muun tyyppiset järjestelmät sisältävät runsaasti tekijöitä, joiden huomioiminen matemaattisissa malleissa ei ole riittävällä tarkkuudella mahdollista. Lisäksi on hyvin todennäköistä, että suurin osa hajautetuista järjestelmistä tullaan rakentamaan tietokoneverkkojen ympärille. Hajauttaminen voidaan periaatteessa suorittaa eriasteisesti riippuen sovelluksesta ja volyymeistä. Ensimmäinen vaihtoehto on suurtehotulostimien hyödyntäminen, jolloin puhutaan usein alueellisesti hajautetusta tulostamisesta, joka ei kuitenkaan perusominaisuuksiltaan merkittävästi poikkea keskitetystä tulostamisesta. Seuraava aste on vähittäisjakeluun hajautettu painatus, jolloin voidaan hyödyntää jo alemman teholuokan tulostimia ja hajautuksen aste on huomattavasti korkeampi. Viimeisenä vaiheena voidaan pitää loppukäyttäjälle hajautettua painatusta.

Mitä pidemmälle hajautusta viedään, sitä vaikeampaa on järjestelmän suorituskyvyn ja kapasiteetin ennustaminen. Suorituskyvyn riittävän tarkka ennustaminen on kuitenkin ehdoton edellytys, jos hajautettua painatusjärjestelmää halutaan hallita ja ohjata tehokkaasti. Suorituskyvyn muodostumiseen vaikuttaa ensinnäkin järjestelmään kuuluvien komponenttien suorituskyyky. Kun hajauttaminen toteutetaan kuten tässä työssä, tietokoneverkon kautta järjestelmä rakentuu seuraavista perusosista:

- Työasema, jolta tulostettava tiedosto lähetetään.
- Tietokoneverkko, jonka kautta tiedosto siirretään.
- Palvelin, joka ohjaa verkon liikennettä ja huolehtii tulostuspalveluista.
- Tulostin, jolla suoritetaan informaation siirto paperille.

Työaseman ominaisuuksilla ei tässä työssä ollut merkittävää vaikutusta muodostuvaan suorituskyykyyn, koska tehtiin oletus että tulostettavasta materiaalista on valmiiksi muodostettu sivunkuvaus. Tätä voidaan tällä hetkellä pitää yleisimpänä käytäntönä eli informatio on tallennettu tietokantaan valmiiksi tulostettavassa muodossa. Huomattavaa



kuitenkin on, että jos informaatiota tulostetaan suoraan sovelluksesta tai esimerkiksi rakenteellisesta tietokannasta, myös sivunkuvauksen muodostamiseen kulunut aika muuttuu järjestelmän suorituskyvyn kannalta oleelliseksi.

Tietokoneverkkojen ominaisuuksilla sen sijaan on suuri merkitys hajautetun tulostusjärjestelmän suorituskyvyn kannalta. Tänä päivänä siirrettävät datamäärät ovat niin mittavia, että varsinkin hajautetun tulostusjärjestelmän kaltaisissa sovelluksissa käytössä oleva siirtokapasiteetti näyttelee merkittävää osaa. Tiedonsiirtotekniikka kehittyy kuitenkin nopeasti ja oletettavaa on, että vähitellen päästään tilanteeseen, jossa tiedonsiirtoon kuluva aika on marginaalinen verrattuna hajautetun tulostusjärjestelmän muihin operaatioihin kuluvaan aikaan. Tämä on mallien luotettavuuden kannalta positiivista, koska verkkokuormituksesta aiheutuvia viiveitä on erittäin vaikea ennustaa ja siten huomioida matemaattisissa malleissa.

Myös palvelimen ominaisuuksilla on vaikutusta järjestelmän kokonaissuorituskykyyn, vaikka merkitys ei ehkä olekaan niin suuri kuin muiden järjestelmän osien. Tämä johtuu siitä, että palvelin on harvoin järjestelmässä pullonkaulatekijä eli muut järjestelmän osat hidastavat prosessia enemmän kuin palvelin. Hajautettu painatusjärjestelmä voi käytännössä sisältää useita palvelimia, jolloin jokainen niistä huolehtii tietyistä niille määrätystä tehtävästä. Yksi tällainen tehtävä voi olla juuri tulostuspalveluiden tarjoaminen verkkoon kytkeytyneille laitteille/käyttäjille eli käytännössä palvelin ohjaa verkon kautta tapahtuvaa tulostamista. Jos järjestelmässä ei ole erillistä tulostinpalvelinta, hoitaa sen tehtäviä yleensä verkon tiedostopalvelin. Laajemmissa MAN- ja WAN-verkoissa joudutaan välttämättä käyttämään useampia palvelimia, koska järjestelmä rakentuu useammista erillisistä lähiverkoista.

Tulostimen ominaisuuksista suorituskyvyn kannalta merkittävänä voidaan pitää seuraavia: koneistonopeus, työmuistin määrä, resoluutio ja rasterointiprosessorin teho. Rasterointiprosessorin tehon määrittäminen on hankalaa, koska rasterointialgoritmin tehokkuutta on vaikea mitata. Tästä syystä tässä työssä päädyttiin suorituskyvyn mittana käyttämään prosessorin kelloaajuutta, joka on helposti selvitettävä parametri. Tämä mitta voi kuitenkin käytännössä osoittautua riittämättömäksi varsinkin silloin jos rasteroinnin toteutuksessa on hyödynnetty useampia prosessoreita tai muuten poikkeavaa prosessoriarkkitehtuuria. Lisäksi tulostinkoneistot saattavat sisältää erilaisia valmistaja-kohtaisia ominaisuuksia, jotka joudutaan mallinnusvaiheessa huomioimaan tapauskohtaisesti.

Järjestelmän ominaisuuksien lisäksi myös tulostettavan materiaalin ominaisuuksilla ja rakenteella on havaittu olevan selvä vaikutus tulostusjärjestelmän läpimenoaikojen muodostumiseen. Merkittäviksi suorituskykyyn vaikuttavista ominaisuuksiksi ovat jo aiemmissa tutkimuksissa osoittautuneet tiedoston koko, sivumäärä, sekä sävykuvien ja viivakuvien määrä. Materiaalin ominaisuudet vaikuttavat suorituskykyyn pääasiassa rasterointiprosessorin välityksellä, koska se on edelleen useimmissa tapauksissa tulostusjärjestelmien pullonkaula. Mitä monimutkaisempaa materiaalia joudutaan käsittelemään, sitä enemmän laskentatehoa prosessorilta vaaditaan ja sitä kauemmin rasterointiin kuluu aikaa. Hajautetun tulostusjärjestelmän yhtenä etuna voidaankin pitää, että rasteroinnin vaatimaa laskentaa saadaan jaettua useammille prosessoreille eli materiaalin käsittelyssä hyödynnetään ikäänkuin rinnakkaislaskentaa.

Käytetyllä sivunkuvauskielillä on merkitystä muodostuvaan suorituskykyyn lähinnä tulostettavan materiaalin sisältämien elementtien kautta. Tämä johtuu jo edellä mainitusta syystä eli mitä monimutkaisempia elementtejä materiaali sisältää, sitä monimutkaisempi on myös muodostettu sivunkuvaus ja sitä enemmän laskentaa sen rasterointi vaatii. Tällä hetkellä on yleisesti käytössä useita sivunkuvauskieliä, joista tässä työssä käsiteltiin ainoastaan Postscriptia sen yleisyyden ja järjestelmäriippumattomuuden takia. Postscript koodin analysointi osoittautui erittäin vaikeaksi, koska sen rakenne riippuu hyvin pitkälle sen generointiin käytetyn sovelluksen tavasta koodata sivunkuvaus. Tämä tarkoittaa, että kahdella eri sovelluksella samasta aineistosta generoidut Postscript-tiedostot saattavat poiketa



rakenteeltaan täysin toisistaan. Tästä syystä mallinnuksessa käytetyn materiaalin rakenne analysoitiin alkuperäisessä muodossaan, ei Postscript-koodina.

Aiemmissa tutkimuksissa on jo havaittu, että elektronisen painatusjärjestelmän suorituskyvyn ennustaminen perinteisillä menetelmillä ei onnistu riittävän tarkasti. Tämän on arveltu johtuvan ainakin kahdesta syystä. Ensinnäkin tulostusjärjestelmään kytkettyjen laitteiden välisten riippuvuuksien ja suorituskykyyn vaikuttavien tekijöiden selvittäminen on ollut vaikeaa. Toiseksi tulostettavan materiaalin sisältämien elementtien yhteisvaikutukset muodostuviin tulostusaikoihin nimenomaan rasterointivaiheessa on havaittu epälineaariseksi. Tämän työn yhteydessä suoritettut testit osoittivat, että tulostusjärjestelmää voidaan kokonaisuutena käsitellä systeeminä, jossa riippuvuudet ovat lineaarisia. Ainoa epälineaarinen osaprosessi on rasterointi, jonka suorituskyvyn ennustaminen lineaarisilla malleilla ei onnistu. Tästä syystä käytettäväksi tulostimien mallinnusmenetelmäksi valittiin neurolaskenta, joka soveltuu erityisesti suuresta datamäärästä vaikeasti havaittavien epälineaaristen riippuvuuksien kuvaamiseen. Neuroverkkojen toiminta perustuu siihen, että niille opetetaan mallinnettavasta ympäristöstä kerätyn esimerkkiaineiston avulla syötteiden ja vasteiden väliset riippuvuudet.

Alunperin tarkoituksena oli mallintaa neuroverkoilla kokonaisia tulostusjärjestelmiä. Tällöin jokaiselle tulostusarkkitehtuurille olisi muodostettu oma neuroverkkomallinsa, joka olisi näin ollen jouduttu myös opettamaan jokaisen arkkitehtuurin tapauksessa erikseen. Tämä lähestymistapa todettiin kuitenkin kömpelöksi, koska järjestelmät sisältävät yleensä runsaasti työasemia, tulostimia ja mahdollisesti myös palvelimia. Näistä voidaan helposti muodostaa lukuisia erilaisia yhdistelmiä, joiden yksittäinen mallintaminen olisi käytännössä erittäin työlästä ja aikaa vievää. Lisäksi päätökseen vaikutti edellä mainittu seikka, että ainoa lineaarisilla malleilla laskettuja ennusteita vääristävä osaprosessi on materiaalin rasterointi, eli muuten lineaarisen järjestelmän mallintaminen runsaasti laskentaa vaativilla neuroverkoilla ei ole järkevää.

Tästä syystä mallintaminen jaettiin kahteen vaiheeseen eli ensin mallinnetaan neuroverkoilla järjestelmän yksittäiset tulostimet, sekä selvitetään muiden järjestelmäkomponenttien suorituskyky niiltä osin kuin se on tarpeellista. Muiden järjestelmäkomponenttien suorituskykyyn vaikuttavat parametrit voidaan yleensä selvittää yksinkertaisilla testituloksilla johtuen niiden lineaarisesta luonteesta. Tämän jälkeen tulostinmallien avulla laskettujen tulostusaikaennusteiden ja selvitettyjen järjestelmäparametrien avulla lasketaan järjestelmän kokonais suorituskyky. Tulostimille neuroverkkomalleja muodostettiin neljä erilaista perusversiota ja kaksi erikoistapauksien testaamiseen. Neljässä perusversiossa varioitiin syötteiden määrää eli käytännössä tulostettavan materiaalin sisältökuvauksen tarkkuutta. Kuvauksen tarkkuuden suhteen voidaan yksiselitteisesti todeta, että mitä tarkempi on neuroverkolle syötetty tulostettavan materiaalin sisältökuvaus, sitä tarkempi on myös mallin tulostusaikaennuste.

Ensimmäinen erikoistapaus oli malli, jossa syötteinä käytettiin ainoastaan tiedoston kokoa, sivumäärää ja EPS-sävykuvien datamäärän osuutta kokonaisdatamäärästä. Nämä syötteet valittiin sillä perusteella, että niillä havaittiin testiaineistoa analysoitaessa selvimmin olevan vaikutusta muodostuviin tulostusaikoihin. Tällä mallilla lasketut tulostusaikaennusteet eivät kuitenkaan olleet yhtä luotettavia kuin tarkimman perusmallin, jossa syötteinä edellä mainittujen kolmen lisäksi käytettiin myös muiden sävykuvien, viivakuvien ja tekstin datamääräosuuksia. Toisessa erikoistapauksessa tarkimman perusmallin syötteisiin lisättiin käytettävän tulostimen suorituskykyä kuvaavia parametrejä. Tämän yleisen tulostinmallin avulla lasketut ennusteet olivat hyvin lähellä tarkimpien tulostinkohtaisten mallien ennusteita. Tästä voidaan päätellä, että on mahdollista muodostaa yleisiä järjestelmäkohtaisia malleja, joiden avulla voidaan ennustaa esimerkiksi järjestelmään kytkettävän uuden tulostimen suorituskykyä.

Muodostettujen neuroverkkomallien tarkkuus oli kuitenkin suhteellisen huono. Syy tähän voi olla käytetyn opetusaineiston rakenteessa tai sitä saattoi olla käytössä myös liian vähän. Todennäköistä on, että tulostettujen tiedostojen sisältämien elementtien välisten

riippuvuuksien yhteisvaikutukset rasteroinnin tehokkuuteen olivat niin monimutkaisia, että niiden selvittäminen ei käytössä olleen opetusaineiston pohjalta ollut mahdollista. Mallien soveltuvuutta tämän tyyppiseen mallintamiseen puoltaa kuitenkin, että tiedostoista, joiden tulostusaikaennusteet poikkesivat todellisista tulostusajoista kaikkein eniten, virhelähteet olivat selvästi osoitettavissa.

Varsinaisen tulostusjärjestelmän mallintamista varten kehitettiin algoritmeja, joiden avulla on mahdollista määrittää mielivaltaisen tulostusarkkitehtuurin suorituskyyky. Tässä diplomityössä käsiteltiin kolme erityyppistä arkkitehtuuria, joista ensimmäinen ei varsinaisesti ole hajautettu: Siinä ainoastaan tulostetaan tiedostoja yhdeltä työasemalta yhdelle tulostimelle. Näin pyrittiin simuloimaan niin sanottua "pipeline"-tulostusarkkitehtuuria. Toisessa testatuista arkkitehtuureista tulostettiin yhdeltä työasemalta samanaikaisesti useammalle tulostimelle ja kolmannessa useammalta työasemalta samanaikaisesti useammalle tulostimelle.

Koska neuroverkkojen avulla muodostetut tulostinmallit eivät olleet riittävän tarkkoja, päädyttiin järjestelmämallien testauksessa käyttämään ennusteiden sijasta mitattuja tulostusaikoja. Näin varmistettiin, että neuroverkkomallien epätarkkuudet eivät vääristä järjestelmämallien ennusteita. Hajautetun painatusjärjestelmän mallien tarkkuus oli testeissä todella hyvä eli virheet olivat keskimäärin alle prosentin luokka. Kannattaa kuitenkin muistaa, että näin tarkkoihin tuloksiin pääseminen edellyttää yksittäisten tulostimien suorituskyyvyn ennustamista riittävän tarkasti. Jatkokehityksen kannalta olisi kuitenkin tärkeää, että kehitetyt neuroverkkomallit ja algoritmit voitaisiin integroida osaksi reaalista tuotantoprosessia. Näin saataisiin kerättyä opetusaineistoa neuroverkoilla muodostettujen tulostinmallien tarkentamiseksi ja samalla pystyttäisiin tutkimaan testiympäristössä kehitettyjen järjestelmäalgoritmien toimivuutta laajemmassa ympäristössä.



# LÄHDELUETTELO

1. **Anon.** *Color, Networks, Large-Format Output*. Seybold Special Report 3(1994)3. ss.3-16.
2. **Anon.** *Digital Color Printing*. Electronic Photography News 9(1995)4. ss. 6-8.
3. **Anon.** *Elektroniseen painoviestintään vauhtia 60 miljoonalla*. Tekniikka & Talous 35(1995)24. s. 4.
4. **Anon.** *Imagesetters, Platesetters and Digital Presses*. Seybold Special Report 3(1995)8. ss. 44-65.
5. **Anon.** *Managing Color, Image Input, Editing and Output; Page Makeup etc*. Seybold Special Report 3(1995)9. ss. 3-15.
6. **Anon.** *Output Issues: Imagesetters, New Film Technology, Computer to Plate, Color Printers*. Seybold Special Report 3(1994)2. ss. 30-67.
7. **Anon.** *PostScript language, Reference Manual*. 7. painos. Reading, Massachusetts, USA 1987. Addison-Wesley Publishing Company, Inc. 322 s.
8. **Anon.** *Telesanasto*. 3. painos. Helsinki 1991. Tekniikan sanastokeskus ry. 156 s.
9. **Antikainen, Hannele & Juntunen, Seppo.** *Elektroninen painaminen. Toimintape-  
riaatteet ja laitteet*. Espoo 1994. VTT, Graafisen tekniikan asiantuntijapalvelu. GT-  
Raportti 4/1994. 20 s.
10. **Autere, Jussi.** *ATM ei vielä sovi Internetin alustaksi*. Tekniikka & Talous 35(1995)18. s. 39.
11. **Bertsekas, Dimitri & Gallager, Robert.** *Data Networks*. Englewood Cliffs, New Jersey 1987. Prentice-Hall, Inc. 486 s.
12. **Davalo, Eric & Naïm, Patrick.** *Neural Networks*. London, England 1990. The MacMillan Press Ltd. 145 s.
13. **Demuth, Howard & Beale, Mark.** *Neural Network Toolbox User's Guide. For Use with MATLAB*. 3. painos. Natick, MA 1994. The Math Works, Inc. 256 s.
14. **Dyson, Peter & Smith, Patricia & Walter, Mark & Tribute, Andrew.** *Output of All Kinds: Eclectic Print News from IPEX*. The Seybold Report on Publishing Systems 23(1993)4. ss. 23-33.
15. **Edwards, Stephen E.** *Linotype-Hell's Hercules: Fast, Versatile Drum Imager to Head IPEX Lineup*. The Seybold Report on Publishing Systems 23(1993)1. ss. 3-10.
16. **Edwards, Stephen E.** *Monotype Systems: New Start for an Old Company*. The Seybold Report on Publishing Systems 23(1993)1. ss. 38-45.
17. **Eliezer, Caren & Edwards, Stephen E.** *Crosfield on Track with Host of New Products*. The Seybold Report on Publishing Systems 23(1994)18. ss. 3-24.

18. **Halsall, Fred.** *Data Communications, Computer Networks and OSI*. 2. painos. Wokingham, England 1990. Addison-Wesley Publishing Company, Inc. 549 s.
19. **Hammond, Joseph L. & O'Reilly, Peter J.P.** *Performance Analysis of Local Computer Networks*. USA 1986. Addison-Wesley Publishing Company, Inc. 411 s.
20. **Haughdahl, J. Scott.** *Inside the Token-Ring*. Amsterdam, Netherlands 1987. Architecture Technology Corporation. 111 s.
21. **Hecht-Nielsen, Robert.** *Neurocomputing*. United States of America 1990. Addison-Wesley Publishing Company, Inc. 433 s.
22. **Hämäläinen, Pertti.** *Token Ring käännepisteessä*. Tietokone (1995)4. ss.81-82.
23. **Isomursu, Pekka & Niskanen, Vesa & Carlsson, Christer & Eklund, Patrik.** *Sumean logiikan mahdollisuudet*. Helsinki 1993. Teknologian kehittämiskeskus (TEKES). Julkaisu nro. 34/93. 100 s.
24. **Kay, David C. & Levine John R.** *Graphics File Formats*. USA 1992. Windcrest/McGraw-Hill. 278 s.
25. **Keränen, Olavi.** *IEEE 802.6-pohjaiset kaupunkiverkot (MAN)*. Nopeat lähialueverkot: LAN, MAN, FDDI ja SMDS. Rantasipi Ikituuri, Turku. 25.-26.3.1992. INSKO Insinöörijärjestöjen koulutuskeskus: Julkaisu 50-92. Osa III ss. 1-16.
26. **Koikkalainen, Pasi.** *Neurocomputing Systems: Formal Modeling and Software Implementation*. Lappeenranta 1992. Lappeenrannan teknillinen korkeakoulu. Tieteellisiä julkaisuja 23. 132 s.
27. **Koikkalainen, Pasi (toim.).** *Neurolaskennan mahdollisuudet*. Helsinki 1994. Teknologian kehittämiskeskus (TEKES). Julkaisu nro. 43/94. 151 s.
28. **Kosko, B.** *Neural Networks and Fuzzy Systems: a dynamical systems approach to machine intelligence*. Englewood Cliffs (NJ) 1992. Prentice-Hall International Inc. 449 s.
29. **Lehtonen, Tapio.** *Painoviestinnän uutta tuotantotekniikkaa*. Painomaailma (1995)1. ss. 44-49.
30. **Lehtonen, Tapio.** *Tietoverkosta painotuotteeksi*. GT-Seminaari: Julkaisuteollisuus ja tietoverkot, Espoo 8.2.1995. Espoo 1995. VTT Tietotekniikka. ss. 1-7.
31. **Levine, Daniel S.** *Introduction to Neural and Cognitive Modeling*. Hillsdale, New Jersey 1991. Lawrence Erlbaum Associates, Inc. 439 s.
32. **Madron, Thomas W.** *Local Area Networks. The Second Generation*. USA 1988. John Wiley & Sons, Inc. 255 s.
33. **Mäkitalo, Pertti.** *Elektronisen painatusprosessin suorituskykyanalyysi*. Espoo, Otaniemi 1994. Teknillinen korkeakoulu T-Aut/Graafinen tekniikka, Diplomityö. 71 s.
34. **Nors, Minna.** *Resoluution vaikutus tulostusjäljen laatuun*. Espoo, Otaniemi 1994. Teknillinen korkeakoulu T-Aut/Graafinen tekniikka, Diplomityö. 138 s.
35. **Oittinen, Pirkko T. & Saarela, Hannu J.** *Elektroninen painaminen*. Espoo 1992. Otatieto Oy. 220 s.



36. **Oittinen, P. & Saarelma, H. & Löppönen, J.-M. & Juhola, H. & Lindqvist, U.** *Elektroninen painoviestintä-tietoverkosta painotuotteeksi (Esiselvitys)*. Espoo 1994. Teknillinen Korkeakoulu, Graafisen tekniikan laboratorio. 93 s.
37. **Palmer-Stevens, David.** *FDDI –A Practical Solution*. Nopeat lähialueverkot: LAN, MAN, FDDI ja SMDS. Rantasipi Ikituuri, Turku. 25.-26.3.1992. INSKO Insinöörijärjestöjen koulutuskeskus: Julkaisu 50-92. Osa II ss. 1-7.
38. **Poropudas, Timo.** *Operatiivinen toiminta on mennyt verkkoon*. Tietoverkko 5(1995)3. s. 36.
39. **Saarelma, Hannu J. & Oittinen Pirkko T.** *Fundamentals of Printing Technology*. 2. painos. Espoo 1994. TKK/Graafisen tekniikan laboratorio. 245 s.
40. **Saari, Susanna.** *Värierottelujen tulostuksen läpimenoaikaan vaikuttavat tekijät DTP-järjestelmässä*. Espoo, Otaniemi 1993. Teknillinen korkeakoulu T-Aut/Graafinen tekniikka, Diplomityö. 81 s.
41. **Stacy, Alan H.** *Interconnectivity. Open Enterprise Networking*. USA 1989. Ungermann-Bass. 214 s.
42. **Stallings, William.** *Data and Computer Communications*. 4. painos. New York 1994. Macmillan Publishing Company. 875 s.
43. **Stallings, William.** *Local Network Technology*. 2. painos. Washington, D.C. 1985. IEEE Computer Society Press. 429 s.
44. **Taylor, J. G. & Mannion, C. L. T (toim.).** *New Developments in Neural Computing*. England 1989. IOP Publishing Ltd. 253 s.
45. **Tenhovuori, Timo.** *Kaupunkiverkot*. Nopeat lähialueverkot: LAN, MAN, FDDI ja SMDS. Rantasipi Ikituuri, Turku. 25.-26.3.1992. INSKO Insinöörijärjestöjen koulutuskeskus: Julkaisu 50-92. Osa VIII ss. 1-5.
46. **Tribute, Andrew.** *Linotype-Hell Unveils Gutenberg Platesetter, New Imagesetters and RIPs*. The Seybold Report on Publishing Systems 24(1995)15. ss. 8-15.
47. **Tribute, Andrew & Edwards, Stephen E.** *Crosfield Changes Focus, Adds Two Imagesetters, Computer-to-Plate Device*. The Seybold Report on Publishing Systems 24(1995)15. ss. 3-7.

# LIITE 1: OPETUSMATERIAALIN RAKENNE JA DATAMÄÄRÄT

Tiedoston nimi	koko [Mt]	Sivu- määrä [kpl]	Fontit/sivu [kpl]	Tekstin datamäärä [kt]	Sävykuvien datamäärä [kt]				Viivakuvien datamäärä [kt]			
					bmp	eps	tiff	TOT	cgm	eps	wmf	TOT
tst001.ps	0,756	24	1	97	0	0	0	0	0	0	0	0
tst002.ps	0,284	3	0	0	0	0	0	0	177	0	0	177
tst003.ps	4,175	3	0	0	3628	0	0	3628	0	0	0	0
tst004.ps	0,515	12	6	28	0	0	0	0	177	0	0	177
tst005.ps	6,976	36	9	83	1513	1808	0	3321	0	0	0	0
tst006.ps	9,760	15	0	0	1513	2838	1808	6159	0	733	39	772
tst007.ps	0,206	9	11	26	0	0	0	0	0	0	0	0
tst008.ps	5,796	21	4	36	1513	0	1328	2841	0	0	103	103
tst009.ps	1,706	15	2	44	163	352	231	747	33	184	0	217
tst010.ps	0,401	12	7	31	75	0	0	75	0	0	9	9
tst011.ps	0,625	30	9	83	0	0	0	0	0	0	0	0
tst012.ps	1,256	6	0	0	0	0	0	0	0	1416	0	1416
tst013.ps	2,699	3	0	0	0	2838	0	2838	0	0	0	0
tst014.ps	1,776	33	1	101	0	0	0	0	192	675	28	895
tst015.ps	8,908	45	3	130	1506	1919	1621	5046	0	0	0	0
tst016.ps	5,938	18	0	0	1506	1919	0	3424	218	733	103	1054
tst017.ps	0,286	9	3	30	0	0	0	0	0	0	0	0
tst018.ps	2,216	9	14	13	863	0	0	863	176	150	0	327
tst019.ps	2,143	24	1	97	218	300	222	739	33	178	0	211
tst020.ps	0,343	12	11	32	0	81	0	81	0	0	8	8
tst021.ps	0,181	9	7	25	0	0	0	0	0	0	0	0
tst022.ps	0,111	6	0	0	0	0	0	0	0	0	103	103
tst023.ps	3,071	3	0	0	0	0	1621	1621	0	0	0	0
tst024.ps	1,429	15	7	25	0	0	0	0	0	1416	0	1416
tst025.ps	11,826	30	1	93	3628	3984	1648	9259	0	0	0	0
tst026.ps	4,087	9	0	0	1621	0	0	1621	218	733	0	951
tst027.ps	0,226	9	4	28	0	0	0	0	0	0	0	0
tst028.ps	3,684	15	8	26	811	0	600	1410	218	381	0	599
tst029.ps	2,228	33	9	97	263	325	220	808	32	176	0	208
tst030.ps	0,448	12	12	33	0	0	74	74	0	0	22	22
tst031.ps	0,160	6	8	19	0	0	0	0	0	0	0	0
tst032.ps	1,028	6	0	0	0	0	0	0	218	733	0	951
tst033.ps	4,822	6	0	0	1506	1919	0	3424	0	0	0	0
tst034.ps	0,636	18	10	35	0	0	0	0	194	0	39	233
tst035.ps	4,249	6	14	9	3628	0	0	3628	0	0	0	0
tst036.ps	8,495	18	0	0	0	3984	1648	5632	194	1416	39	1649
tst037.ps	0,310	9	1	29	0	0	0	0	0	0	0	0
tst038.ps	0,951	6	14	11	0	800	0	800	65	0	0	65
tst039.ps	1,826	39	3	110	253	81	236	570	33	178	0	211
tst040.ps	0,573	9	8	22	163	0	0	163	0	0	47	47
tst041.ps	0,281	15	2	44	0	0	0	0	0	0	0	0
tst042.ps	0,357	6	0	0	0	0	0	0	194	0	39	233
tst043.ps	6,376	6	0	0	1513	0	1808	3321	0	0	0	0
tst044.ps	1,642	39	9	87	0	0	0	0	177	733	39	950
tst045.ps	3,449	12	6	27	534	1416	394	2344	0	0	0	0
tst046.ps	2,799	9	0	0	0	2838	0	2838	0	0	103	103
tst047.ps	0,232	9	12	28	0	0	0	0	0	0	0	0
tst048.ps	3,109	12	8	18	0	2838	0	2838	177	0	0	177
tst049.ps	2,116	18	4	52	247	313	217	776	32	430	0	462
tst050.ps	0,459	9	3	22	0	0	190	190	0	0	22	22



LIITE 2: OPETUSMATERIAALIN RAKENNE (%-OSUUDET)

Tiedoston nimi	Tekstin osuus [%]	Sävykuvien osuus [%]				Viivakuvien osuus [%]			
		bmp	eps	tiff	TOT	cgm	eps	wmf	TOT
tst001.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst002.ps	0 %	0 %	0 %	0 %	0 %	100 %	0 %	0 %	100 %
tst003.ps	0 %	100 %	0 %	0 %	100 %	0 %	0 %	0 %	0 %
tst004.ps	14 %	0 %	0 %	0 %	0 %	86 %	0 %	0 %	86 %
tst005.ps	2 %	44 %	53 %	0 %	98 %	0 %	0 %	0 %	0 %
tst006.ps	0 %	22 %	41 %	26 %	89 %	0 %	11 %	1 %	11 %
tst007.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst008.ps	1 %	51 %	0 %	45 %	95 %	0 %	0 %	3 %	3 %
tst009.ps	4 %	16 %	35 %	23 %	74 %	3 %	18 %	0 %	22 %
tst010.ps	27 %	65 %	0 %	0 %	65 %	0 %	0 %	8 %	8 %
tst011.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst012.ps	0 %	0 %	0 %	0 %	0 %	0 %	100 %	0 %	100 %
tst013.ps	0 %	0 %	100 %	0 %	100 %	0 %	0 %	0 %	0 %
tst014.ps	10 %	0 %	0 %	0 %	0 %	19 %	68 %	3 %	90 %
tst015.ps	3 %	29 %	37 %	31 %	97 %	0 %	0 %	0 %	0 %
tst016.ps	0 %	34 %	43 %	0 %	76 %	5 %	16 %	2 %	24 %
tst017.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst018.ps	1 %	72 %	0 %	0 %	72 %	15 %	13 %	0 %	27 %
tst019.ps	9 %	21 %	29 %	21 %	71 %	3 %	17 %	0 %	20 %
tst020.ps	27 %	0 %	67 %	0 %	67 %	0 %	0 %	7 %	7 %
tst021.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst022.ps	0 %	0 %	0 %	0 %	0 %	0 %	0 %	100 %	100 %
tst023.ps	0 %	0 %	0 %	100 %	100 %	0 %	0 %	0 %	0 %
tst024.ps	2 %	0 %	0 %	0 %	0 %	0 %	98 %	0 %	98 %
tst025.ps	1 %	39 %	43 %	18 %	99 %	0 %	0 %	0 %	0 %
tst026.ps	0 %	63 %	0 %	0 %	63 %	8 %	29 %	0 %	37 %
tst027.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst028.ps	1 %	40 %	0 %	29 %	69 %	11 %	19 %	0 %	29 %
tst029.ps	9 %	24 %	29 %	20 %	73 %	3 %	16 %	0 %	19 %
tst030.ps	25 %	0 %	0 %	58 %	58 %	0 %	0 %	17 %	17 %
tst031.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst032.ps	0 %	0 %	0 %	0 %	0 %	23 %	77 %	0 %	100 %
tst033.ps	0 %	44 %	56 %	0 %	100 %	0 %	0 %	0 %	0 %
tst034.ps	13 %	0 %	0 %	0 %	0 %	72 %	0 %	14 %	87 %
tst035.ps	0 %	100 %	0 %	0 %	100 %	0 %	0 %	0 %	0 %
tst036.ps	0 %	0 %	55 %	23 %	77 %	3 %	19 %	1 %	23 %
tst037.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst038.ps	1 %	0 %	91 %	0 %	91 %	7 %	0 %	0 %	7 %
tst039.ps	12 %	28 %	9 %	27 %	64 %	4 %	20 %	0 %	24 %
tst040.ps	9 %	70 %	0 %	0 %	70 %	0 %	0 %	20 %	20 %
tst041.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst042.ps	0 %	0 %	0 %	0 %	0 %	83 %	0 %	17 %	100 %
tst043.ps	0 %	46 %	0 %	54 %	100 %	0 %	0 %	0 %	0 %
tst044.ps	8 %	0 %	0 %	0 %	0 %	17 %	71 %	4 %	92 %
tst045.ps	1 %	23 %	60 %	17 %	99 %	0 %	0 %	0 %	0 %
tst046.ps	0 %	0 %	96 %	0 %	96 %	0 %	0 %	4 %	4 %
tst047.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
tst048.ps	1 %	0 %	94 %	0 %	94 %	6 %	0 %	0 %	6 %
tst049.ps	4 %	19 %	24 %	17 %	60 %	2 %	33 %	0 %	36 %
tst050.ps	9 %	0 %	0 %	81 %	81 %	0 %	0 %	9 %	9 %

LIITE 3: TESTIMATERIAALIN RAKENNE

Taulukko 1. Testiaineistona käytetyn materiaalin rakenne ja sen sisältämien elementtien datamäärät.

Tiedoston nimi	koko [Mt]	Sivu- määrä [kpl]	Fontit/sivu [kpl]	Tekstin datamäärä [kt]	Sävykuvien datamäärä				Viivakuvien datamäärä			
					[kt]				[kt]			
					bmp	eps	tiff	TOT	cgm	eps	wmf	TOT
wmfeps.ps	18,613	136	–	255	0	7507	0	7763	0	0	4166	4166
epseps.ps	12,166	136	–	255	0	7507	0	7763	0	2605	0	2605
cgmeeps.ps	12,503	136	–	255	0	7507	0	7763	3241	0	0	3241
epsbmp.ps	12,101	136	–	255	6743	0	0	6999	0	2605	0	2605
epstiff.ps	12,100	136	–	255	0	0	3641	3897	0	2605	0	2605
grakbmp.ps	5,076	32	–	59	2034	0	0	2092	0	0	0	0
grakeps.ps	5,237	32	–	59	0	4231	0	4289	0	0	0	0
graktiff.ps	5,076	32	–	59	0	0	3909	3967	0	0	0	0
opetus_o.ps	4,107	64	–	354	0	0	0	354	0	0	0	0

Taulukko 2. Taulukossa 1 esitettyjen elementtien %-osuudet kokonaisdatamäärästä.

Tiedoston nimi	Tekstin osuus [%]	Sävykuvien osuus				Viivakuvien osuus			
		[%]				[%]			
		bmp	eps	tiff	TOT	cgm	eps	wmf	TOT
wmfeps.ps	2 %	0 %	63 %	0 %	63 %	0 %	0 %	35 %	35 %
epseps.ps	2 %	0 %	72 %	0 %	72 %	0 %	25 %	0 %	25 %
cgmeeps.ps	2 %	0 %	68 %	0 %	68 %	29 %	0 %	0 %	29 %
epsbmp.ps	3 %	70 %	0 %	0 %	70 %	0 %	27 %	0 %	27 %
epstiff.ps	4 %	0 %	0 %	56 %	56 %	0 %	40 %	0 %	40 %
grakbmp.ps	3 %	97 %	0 %	0 %	97 %	0 %	0 %	0 %	0 %
grakeps.ps	1 %	0 %	99 %	0 %	99 %	0 %	0 %	0 %	0 %
graktiff.ps	1 %	0 %	0 %	99 %	99 %	0 %	0 %	0 %	0 %
opetus_o.ps	100 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %



## LIITE 4: OPETUSMATERIAALIN SIIRTO- JA TULOSTUSAJAT

Tiedoston nimi	Siirtoaika [s]	HP LaserJet 4/4M+		IBM LaserPrinter 4029 PS17		Canon CLC 500+ Fiery 125i	
		Tulostusaika [s]	Yhteensä [s]	Tulostusaika [s]	Yhteensä [s]	Tulostusaika [s]	Yhteensä [s]
tst001.ps	36	176	212	376	412	769	805
tst002.ps	14	42	56	123	137	101	115
tst003.ps	200	1764	1964	4877	5077	1275	1475
tst004.ps	25	95	120	234	259	384	409
tst005.ps	334	2228	2562	6019	6353	2543	2877
tst006.ps	467	2264	2731	6068	6535	2036	2503
tst007.ps	10	69	79	137	147	282	292
tst008.ps	278	1971	2249	5148	5426	1931	2209
tst009.ps	82	409	491	990	1072	692	774
tst010.ps	19	119	138	268	287	392	411
tst011.ps	30	172	202	314	344	905	935
tst012.ps	60	110	170	304	364	214	274
tst013.ps	129	124	253	215	344	225	354
tst014.ps	85	270	355	623	708	1082	1167
tst015.ps	427	2437	2864	6168	6595	2933	3360
tst016.ps	284	1300	1584	3324	3608	1414	1698
tst017.ps	14	77	91	155	169	290	304
tst018.ps	106	662	768	1849	1955	709	815
tst019.ps	103	484	587	1159	1262	987	1090
tst020.ps	16	87	103	155	171	377	393
tst021.ps	9	66	75	119	128	270	279
tst022.ps	5	49	54	99	104	178	183
tst023.ps	147	1045	1192	2757	2904	802	949
tst024.ps	68	142	210	380	448	484	552
tst025.ps	566	3148	3714	8310	8876	3053	3619
tst026.ps	196	1119	1315	2984	3180	1003	1199
tst027.ps	11	65	76	140	151	282	293
tst028.ps	176	1048	1224	2766	2942	1120	1296
tst029.ps	107	533	640	1283	1390	1274	1381
tst030.ps	21	134	155	323	344	421	442
tst031.ps	8	51	59	107	115	199	207
tst032.ps	49	95	144	279	328	208	257
tst033.ps	231	1187	1418	3009	3240	1013	1244
tst034.ps	30	121	151	283	313	563	593
tst035.ps	203	1782	1985	4930	5133	1374	1577
tst036.ps	407	1297	1704	3527	3934	1429	1836
tst037.ps	15	78	93	156	171	295	310
tst038.ps	46	91	137	200	246	238	284
tst039.ps	87	529	616	1220	1307	1371	1458
tst040.ps	27	183	210	473	500	362	389
tst041.ps	13	95	108	157	170	448	461
tst042.ps	17	58	75	155	172	186	203
tst043.ps	305	2175	2480	5755	6060	1647	1952
tst044.ps	79	256	335	546	625	1200	1279
tst045.ps	165	747	912	1852	2017	868	1033
tst046.ps	134	157	291	292	426	391	525
tst047.ps	11	71	82	135	146	290	301
tst048.ps	149	183	332	357	506	496	645
tst049.ps	101	508	609	1227	1328	853	954
tst050.ps	22	179	201	428	450	346	368

LIITE 5: HAJAUTETTUIJEN JÄRJESTELMIEN TESTIEN TULOKSET

Taulukko 1. HP LasetJet 4/4M+ PS:n datasiirtonopeudet ja tulostusajat “pipeline”-simulaati-  
ossa.

		Siirto tot [min:s]	Siirto tot [s]	Siirto/tiedosto [s]	Koko [kt]	Siirtonopeus [kt/s]	Tulostusaika [min:s]	Tulostusaika [s]
Testisarja I								
1.	tst002.ps	00 : 14	14	14	284	20,29	13 : 10	790
2.	tst018.ps	02 : 00	120	106	2216	20,91		
Testisarja II								
1.	tst001.ps	00 : 37	37	37	756	20,43	35 : 49	2149
2.	tst003.ps	03 : 56	236	199	4175	20,98		
3.	tst001.ps	04 : 32	272	36	756	21,00		
Testisarja III								
1.	tst009.ps	01 : 22	82	82	1706	20,80	27 : 49	1669
2.	tst009.ps	02 : 43	163	81	1706	21,06		
3.	tst009.ps	04 : 04	244	81	1706	21,06		
4.	tst009.ps	05 : 24	324	80	1706	21,33		
Testisarja IV								
1.	tst010.ps	00 : 20	20	20	401	20,05	09 : 16	556
2.	tst010.ps	00 : 39	39	19	401	21,11		
3.	tst010.ps	00 : 58	58	19	401	21,11		
4.	tst010.ps	01 : 17	77	19	401	21,11		
5.	tst010.ps	01 : 36	96	19	401	21,11		
Testisarja V								
1.	tst002.ps	00 : 14	14	14	284	20,29	03 : 14	194
2.	tst002.ps	00 : 28	28	14	284	20,29		
3.	tst002.ps	00 : 41	41	13	284	21,85		
4.	tst002.ps	00 : 55	55	14	284	20,29		
5.	tst002.ps	01 : 08	68	13	284	21,85		
6.	tst002.ps	01 : 22	82	14	284	20,29		



**Taulukko 2.** IBM LaserPrinter 4029 PS17 datasiirtonopeudet ja tulostusajat "pipeline"-simulaatiossa.

		Siirto tot [min:s]	Siirto tot [s]	Siirto/tiedosto [s]	Koko [kt]	Siirtonopeus [kt/s]	Tulostusaika [min:s]	Tulostusaika [s]
Testisarja I								
1.	tst002.ps	00 : 14	14	14	284	20,29	32 : 56	1976
2.	tst018.ps	02 : 00	120	106	2216	20,91		
Testisarja II								
1.	tst001.ps	00 : 37	37	37	756	20,43	92 : 31	5551
2.	tst003.ps	03 : 55	235	198	4175	21,09		
3.	tst001.ps	04 : 31	271	36	756	21,00		
Testisarja III								
1.	tst009.ps	01 : 21	81	81	1706	21,06	66 : 09	3969
2.	tst009.ps	02 : 43	163	82	1706	20,80		
3.	tst009.ps	04 : 03	243	80	1706	21,33		
4.	tst009.ps	05 : 23	323	80	1706	21,33		
Testisarja IV								
1.	tst010.ps	00 : 20	20	20	401	20,05	21 : 16	1276
2.	tst010.ps	00 : 38	38	18	401	22,28		
3.	tst010.ps	00 : 57	57	19	401	21,11		
4.	tst010.ps	01 : 16	76	19	401	21,11		
5.	tst010.ps	01 : 35	95	19	401	21,11		
Testisarja V								
1.	tst002.ps	00 : 14	14	14	284	20,29	10 : 35	635
2.	tst002.ps	00 : 28	28	14	284	20,29		
3.	tst002.ps	00 : 42	42	14	284	20,29		
4.	tst002.ps	00 : 54	54	12	284	23,67		
5.	tst002.ps	01 : 08	68	14	284	20,29		
6.	tst002.ps	01 : 21	81	13	284	21,85		

**Taulukko 3.** Canon CLC 500+Fiery 125i:n datasiirtonopeudet ja tulostusajat “pipeline”-simulaatiossa.

		Siirto tot [min:s]	Siirto tot [s]	Siirto/tiedosto [s]	Koko [kt]	Siirtonopeus [kt/s]	Tulostusaika [min:s]	Tulostusaika [s]
Testisarja I								
1.	tst002.ps	00 : 14	14	14	284	20,29	13 : 45	825
2.	tst018.ps	02 : 00	120	106	2216	20,91		
Testisarja II								
1.	tst001.ps	00 : 36	36	36	756	21,00	46 : 40	2800
2.	tst003.ps	03 : 56	236	200	4175	20,88		
3.	tst001.ps	04 : 35	275	39	756	19,38		
Testisarja III								
1.	tst009.ps	01 : 22	82	82	1706	20,80	46 : 36	2796
2.	tst009.ps	02 : 44	164	82	1706	20,80		
3.	tst009.ps	04 : 06	246	82	1706	20,80		
4.	tst009.ps	05 : 28	328	82	1706	20,80		
Testisarja IV								
1.	tst010.ps	00 : 19	19	19	401	21,11	32 : 39	1959
2.	tst010.ps	00 : 38	38	19	401	21,11		
3.	tst010.ps	00 : 58	58	20	401	20,05		
4.	tst010.ps	01 : 17	77	19	401	21,11		
5.	tst010.ps	01 : 36	96	19	401	21,11		
Testisarja V								
1.	tst002.ps	00 : 14	14	14	284	20,29	10 : 04	604
2.	tst002.ps	00 : 28	28	14	284	20,29		
3.	tst002.ps	00 : 41	41	13	284	21,85		
4.	tst002.ps	00 : 55	55	14	284	20,29		
5.	tst002.ps	01 : 09	69	14	284	20,29		
6.	tst002.ps	01 : 22	82	13	284	21,85		



**Taulukko 4.** Yhden työaseman ja useamman tulostimen arkkitehtuurien datasiirtonopeudet ja tulostusajat.

		Tulostin	Siirto		Siirto/tiedosto	Koko	Siirtonopeus	Tulostusaika	
			[min:s]	[s]	[s]	[kt]	[kt/s]	[min:s]	[s]
Testisarja I									
1.	tst002.ps	IBM	00 : 14	14	14	284	20,29	02 : 15	135
2.	tst002.ps	Canon	00 : 28	28	14	284	20,29		
3.	tst002.ps	HP	00 : 42	42	14	284	20,29		
Testisarja II									
1.	tst009.ps	IBM	01 : 21	81	81	1706	21,06	18 : 01	1081
2.	tst009.ps	Canon	02 : 42	162	81	1706	21,06		
3.	tst009.ps	HP	04 : 04	244	82	1706	20,80		
Testisarja III									
1.	tst001.ps	Canon	00 : 37	37	37	756	20,43	13 : 20	800
2.	tst001.ps	IBM	01 : 13	73	36	756	21,00		
3.	tst001.ps	HP	01 : 48	108	35	756	21,60		
Testisarja IV									
1.	tst010.ps	Canon	00 : 20	20	20	401	20,05	06 : 52	412
2.	tst010.ps	IBM	00 : 39	39	19	401	21,11		
3.	tst010.ps	HP	00 : 59	59	20	401	20,05		
Testisarja V									
1.	tst003.ps	IBM	03 : 20	200	200	4175	20,88	84 : 32	5072
2.	tst003.ps	HP	06 : 39	399	199	4175	20,98		
3.	tst003.ps	Canon	09 : 58	598	199	4175	20,98		
Testisarja VI									
1.	tst012.ps	IBM	01 : 01	61	61	1256	20,59	27 : 13	1633
2.	tst016.ps	HP	05 : 45	345	284	5938	20,91		
3.	tst019.ps	Canon	07 : 28	448	103	2143	20,81		
Testisarja VII									
1.	tst018.ps	HP	01 : 47	107	107	2216	20,71	12 : 54	774
2.	tst013.ps	Canon	03 : 57	237	130	2699	20,76		
3.	tst020.ps	IBM	04 : 14	254	17	343	20,18		
Testisarja VIII									
1.	tst017.ps	Canon	00 : 14	14	14	286	20,43	12 : 15	735
2.	tst014.ps	IBM	01 : 39	99	85	1776	20,89		
3.	tst019.ps	HP	03 : 22	202	103	2143	20,81		
Testisarja IX									
1.	tst012.ps	Canon	01 : 01	61	61	1256	20,59	4 : 35	275
2.	tst011.ps	HP	01 : 32	92	31	625	20,16		
Testisarja X									
1.	tst017.ps	HP	00 : 14	14	14	286	20,43	05 : 09	309
2.	tst010.ps	IBM	00 : 34	34	20	401	20,05		

**Taulukko 5.** Useamman työasemasta ja tulostimesta muodostuvien arkkitehturien mitatut data-siirtonopeudet ja tulostusajat.

		Työ- asema	Tulostin	Siirto		Koko [kt]	Siirtonopeus [kt/s]	Tulostusaika	
				[min:s]	[s]			[min:s]	[s]
Testisarja I									
1.	tst022.ps	2	HP	00 : 07	7	111	15,86	22 : 42	1362
2.	tst024.ps	3	IBM	01 : 28	88	1429	16,24		
3.	tst033.ps	4	Canon	05 : 48	348	4822	13,86		
Testisarja II									
1.	tst032.ps	2	IBM	01 : 03	63	1028	16,32	05 : 47	347
2.	tst022.ps	3	Canon	00 : 07	7	111	15,86		
3.	tst021.ps	4	HP	00 : 13	13	181	13,92		
Testisarja III									
1.	tst023.ps	2	Canon	03 : 06	186	3071	16,51	16 : 17	977
2.	tst027.ps	3	HP	00 : 15	15	226	15,07		
3.	tst040.ps	4	IBM	00 : 45	45	573	12,73		
Testisarja IV									
1.	tst030.ps	2	IBM	00 : 28	28	448	16,00	22 : 43	1363
2.	tst026.ps	3	HP	04 : 05	245	4087	16,68		
3.	tst021.ps	4	Canon	00 : 14	14	181	12,93		
Testisarja V									
1.	tst023.ps	2	HP	03 : 08	188	3071	16,34	20 : 32	1232
2.	tst032.ps	3	Canon	01 : 05	65	1028	15,82		
3.	tst021.ps	4	IBM	00 : 16	16	181	11,31		
Testisarja VI									
1.	tst033.ps	2	Canon	04 : 50	290	4822	16,63	21 : 46	1306
2.	tst024.ps	3	IBM	01 : 28	88	1429	16,24		
3.	tst030.ps	4	HP	00 : 35	35	448	12,80		
Testisarja VII									
1.	tst034.ps	2	HP	00 : 38	38	636	16,74	02 : 39	159
2.	tst031.ps	3	IBM	00 : 13	13	160	12,31		
Testisarja VIII									
1.	tst024.ps	3	HP	01 : 25	85	1429	16,81	05 : 08	308
2.	tst038.ps	4	Canon	01 : 09	69	951	13,78		
Testisarja IX									
1.	tst030.ps	2	IBM	00 : 28	28	448	16,00	06 : 43	403
2.	tst040.ps	4	Canon	00 : 43	43	573	13,33		
Testisarja X									
1.	tst040.ps	2	IBM	00 : 36	36	573	15,92	20 : 27	1227
2.	tst023.ps	3	HP	03 : 03	183	3071	16,78		



**LIITE 6: SUORITUSKYKYALGORITMIEN C-LÄHDEKODI**

```
#include <stdio.h>
#include <stdlib.h>

#define asema1_nopeus 20.88
#define asema2_nopeus 16.44
#define asema3_nopeus 16.44
#define asema4_nopeus 13.97

#define viive_hp 14.80
#define viive_ibm 22.99
#define viive_fiery 4.12

int tuotetiedot (void)
{
    int tuote;

    printf("\nTulostettavan materiaalin rakenne ?\n");
    printf("1 - Useampia kappaleita samasta tiedostosta.\n");
    printf("2 - Useita sisällöltään erilaisia tiedostoja.\n");
    scanf("%i",&tuote);
    return tuote;
}

float valitse_tulostin (void)
{
    int t;
    float v;

    printf("\nValitse käytettävä tulostin\n");
    printf("1 - HP LaserJet 4/4M+ PS\n");
    printf("2 - IBM LaserPrinter 4029 PS17\n");
    printf("3 - Canon CLC 500 + Fiery 125i\n");
    scanf("%i",&t);
    switch (t)
    {
        case 1 :
            v = viive_hp;
            break;
        case 2 :
            v = viive_ibm;
            break;
        case 3 :
            v = viive_fiery;
            break;
    }
    return v;
}

float valitse_asema (void)
{
    int t;
    float n;

    printf("\nValitse käytettävä työasema\n");
    printf("1 - Työasema 1\n");
    printf("2 - Työasema 2\n");
    printf("3 - Työasema 3\n");
    printf("4 - Työasema 4\n");
    scanf("%i",&t);
    switch (t)
    {
        case 1 :
            n = asema1_nopeus;
            break;
```

```

        case 2 :
            n = asema2_nopeus;
            break;
        case 3 :
            n = asema3_nopeus;
            break;
        case 4 :
            n = asema4_nopeus;
            break;
    }
    return n;
}

float one_to_one (void)
{
    int tuotteet,tiedostojen_lkm,ennustettu_tulostusaika,tiedoston_koko,laskuri;
    float nopeus,viive,siirtoaika,tulostus_edellinen,tulostus_seuraava,aika_tot=0;

    tuotteet = tuotetiedot();
    nopeus = valitse_asema();
    viive = valitse_tulostin();
    printf("\nTulostettavien tiedostojen lukumäärä [kpl]? ");
    scanf("%i",&tiedostojen_lkm);
    if (tuotteet == 1)
    {
        printf("\nYhden tiedoston tulostusaika [s]? ");
        scanf("%i",&ennustettu_tulostusaika);
        printf("Tiedoston koko [kt]? ");
        scanf("%i",&tiedoston_koko);
        siirtoaika = tiedoston_koko/nopeus;
        aika_tot = siirtoaika + (tiedostojen_lkm*ennustettu_tulostusaika) -
            ((tiedostojen_lkm-1)*viive);
    }
    else
        for (laskuri = 1;laskuri <= tiedostojen_lkm; laskuri++)
        {
            printf("\ni.:n tiedoston tulostusaika [s]? ",laskuri);
            scanf("%i",&ennustettu_tulostusaika);
            printf("Tiedoston koko [kt]? ");
            scanf("%i",&tiedoston_koko);
            siirtoaika = tiedoston_koko/nopeus;
            if (laskuri == 1)
            {
                aika_tot = siirtoaika + ennustettu_tulostusaika;
                tulostus_edellinen = ennustettu_tulostusaika;
            }
            else
            {
                if (siirtoaika >= tulostus_edellinen)
                    tulostus_seuraava = ennustettu_tulostusaika +
                        (siirtoaika - tulostus_edellinen);
                else
                {
                    tulostus_seuraava = ennustettu_tulostusaika - viive;
                }
                aika_tot += tulostus_seuraava;
                tulostus_edellinen = tulostus_seuraava;
            }
        }
    return aika_tot;
}

float one_to_N (void)
{
    int tuotteet,tulostimien_lkm,laskuri,arvioitu_tulostusaika,tiedoston_koko;
    float nopeus,siirtoaika,siirtoaika_tot=0,aika,aika_tot=0;

```



```

    tuotteet = tuotetiedot();
    nopeus = valitse_asema();
    printf("\nMontako tulostinta [kpl]? ");
    scanf("%i",&tulostimien_lkm);
    if (tuotteet == 1)
    {
        printf("Tiedoston koko [kt]? ");
        scanf("%i",&tiedoston_koko);
        for (laskuri=1;laskuri <= tulostimien_lkm; laskuri++)
        {
            printf("%i.:n tulostimen arvioitu tulostusaika [s]? ",laskuri);
            scanf("%i",&arvioitu_tulostusaika);
            siirtoaika = (tiedoston_koko/nopeus);
            if (laskuri == 1)
                aika_tot = siirtoaika + arvioitu_tulostusaika;
            else
            {
                aika = (siirtoaika + arvioitu_tulostusaika) + siirtoaika_tot;
                if (aika >= aika_tot)
                    aika_tot = aika;
            }
            siirtoaika_tot += siirtoaika;
        }
    }
    else
        for (laskuri=1;laskuri <= tulostimien_lkm; laskuri++)
        {
            printf("%i.:n tulostettavan tiedoston koko [kt]? ",laskuri);
            scanf("%i",&tiedoston_koko);
            printf("%i.:n tulostimen arvioitu tulostusaika [s]? ",laskuri);
            scanf("%i",&arvioitu_tulostusaika);
            siirtoaika = (tiedoston_koko/nopeus);
            if (laskuri == 1)
                aika_tot = siirtoaika + arvioitu_tulostusaika;
            else
            {
                aika = (siirtoaika + arvioitu_tulostusaika)+siirtoaika_tot;
                if (aika >= aika_tot)
                    aika_tot = aika;
            }
            siirtoaika_tot += siirtoaika;
        }
    return aika_tot;
}

float N_to_N (void)
{
    int tuotteet=0, asemien_lkm,laskuri,tiedoston_koko,arvioitu_tulostusaika;
    float nopeus,aika,aika_tot=0;

    tuotteet = tuotetiedot();
    printf("\nMontako lähettävää työasemaa [kpl]? ");
    scanf("%i",&asemien_lkm);
    if (tuotteet == 1)
    {
        printf("Tiedoston koko [kt]? ");
        scanf("%i",&tiedoston_koko);
        for (laskuri=1;laskuri <= asemien_lkm;laskuri++)
        {
            nopeus = valitse_asema();
            printf("%i.:n tulostimen arvioitu tulostusaika [s]? ",laskuri);
            scanf("%i",&arvioitu_tulostusaika);
            aika = arvioitu_tulostusaika + (tiedoston_koko/nopeus);
            if (laskuri == 1)
                aika_tot = aika;
            else
                if (aika >= aika_tot)

```

```
        aika_tot = aika;
    }
}
else
{
    for (laskuri=1;laskuri <= asemien_lkm;laskuri++)
    {
        nopeus = valitse_asema();
        printf("Tiedoston koko [kt]? ");
        scanf("%i",&tiedoston_koko);
        printf("%i.:n tulostimen arvioitu tulostusaika [s]? ",laskuri);
        scanf("%i",&arvioitu_tulostusaika);
        aika = arvioitu_tulostusaika + (tiedoston_koko/nopeus);
        if (laskuri == 1)
            aika_tot = aika;
        else
            if (aika >= aika_tot)
                aika_tot = aika;
    }
}
return aika_tot;
}

int main (void)
{
    int malli;
    float tulostusaika;
    div_t aika;

    printf("Valitse haluamasi arkkitehtuuri\n");
    printf("1 - Yhdeltä työasemalta yhdelle tulostimelle\n");
    printf("2 - Yhdeltä työasemalta useammalle tulostimelle\n");
    printf("3 - Useammalta työasemalta useammalle tulostimelle\n");
    scanf("%i",&malli);
    switch (malli)
    {
        case 1 :
            tulostusaika = one_to_one();
            break;
        case 2 :
            tulostusaika = one_to_N();
            break;
        case 3 :
            tulostusaika = N_to_N();
            break;
    }
    aika = div (tulostusaika, 60);
    printf("\nArvioitu kokonaistulostusaika on %.2f sekuntia\n",tulostusaika);
    printf("eli %i minuuttia %i sekuntia.\n",aika.quot,aika.rem);
    return 0;
}
```



TEKNILLINEN KORKEAKOULU  
TIETOTEKNIKAN TALON KILJATO  
KONEMIEHENTIE 2  
02150 ESPOO

TEKNILLINEN KORKEAKOULU  
Graafisen tekniikan  
laboratorio